

SZAKDOLGOZAT

BUDAPESTI GAZDASÁGI EGYETEM
KÜLKERESKEDELMI KAR
NEMZETKÖZI GAZDÁLKODÁS SZAK
NAPPALI MUNKAREND
ÜZLETI INTELLIGENCIA ÉS DÖNTÉSEK SPECIALIZÁCIÓ

EGY VÁLLALATI ADATMODELL KIÉPÍTÉSE

Belső konzulens: Dr. Budai László

Külső konzulens: Korhut Tamás

Készítette: Varga Kristóf

Budapest, 2024

BEVEZETÉS.....	4
1. A FIKTÍV VÁLLALAT.....	6
1.1. Információ gyűjtés és kutatás	6
1.2. Miért van szüksége egy vállalatnak egy adatbázisra?.....	6
1.3. Egy fiktív vállalat felépítése.....	8
2. ADATBÁZISOK FELÉPÍTÉSE.....	10
2.1. Az adat és az adatbázis.....	10
2.2. A relációs adatbázis modell.....	11
2.3. További adatbázis modellek.....	13
2.4. Az SGM, valamint annak jelölései és szabályai.....	15
2.5. Az adatbázisok normalizációja (NF).....	17
3. ÚJÍTÁSOK ÉS TRENDEK, MI VÁR RÁNK A JÖVŐBEN	18
3.1. Technológiai fejlődés az adatbázisok területén.....	18
3.2. Új trendek és lehetőségek	18
3.3. Az adat, mint termék és a termék, mint adat	22
4. AZ ADATOK ÉS ADATBÁZISOK BIZTONSÁGA.....	26
4.1. Mi az adatbiztonság és miért fontos számunkra?	26
4.2. Hogyan védhetjük adatainkat?.....	26
4.3. Az adatbiztonság gyakorlatai.....	28
5. A FEJLESZTÉS.....	30
5.1. Az elképzelés	30
5.2. A vállalat SGM táblája.....	30
5.3. A program és annak nyelveze	32
5.4. Táblák, kapcsolatok és kulcsok hozzárendelése az adatbázishoz	34
5.5. Táblák feltöltése adatokkal Mesterséges Intelligencia segítségével.....	35
5.6. Lekérdezések és működőképesség.....	38
5.7. Késztermék	40
6. ÖSSZEGZÉS.....	42
Irodalomjegyzék	45
Mellékletek.....	47

BEVEZETÉS

A 21. században a technika előrehaladtával a cégek irányításába is sikerült beépíteni számítástechnikai eszközöket, amik megkönnyíthetik egy vállalat mindennapjait. Egy cég életében nagyon fontos, hogy egy jól átlátható adminisztrációval rendelkezzen, ahol minden olyan információ megtalálható, ami nélkülözhetetlen a zökkenőmentes működéshez.

Korábban minden szükséges információt papíron vezettek, akár az üzletekben az áruk típusáról, darabszámáról, akár logisztikai hálózatoknál a küldemények nyomon követéséről beszélünk. Számtalan példát lehetne hozni arra, hogy hányféle adminisztrációt kell egymással szinkronban működtetni, ami lássuk be papír alapú rendszerrel manapság már szinte lehetetlen. A számítástechnika segítségével egy adatbázison keresztül sokkal megbízhatóbb információkhoz juthatunk, mint egy emberi kéz által vezetett papírtömbbel. Egy ilyen rendszer segítségével könnyebben tudunk az adatbázisban lévő adatokból lekérdezni árukészletet, adott csomagok hollétét, vagy akár automatizálhatunk is általa bizonyos folyamatokat, amik meggyorsítják és megkönnyítik az irányítást.

Miért is választottam ezt a témát? Ebben az évtizedben rendkívül nagy hangsúlyt kellene a nagy vállalatoknak fektetni az adatbázisokra, hiszen a mesterséges intelligencia megjelenésével és a technológia fejlődésével olyan lehetőségek nyílnak meg számukra, amikkel képesek lehetnek költséget csökkenteni, árbevételt maximalizálni és az üzleti életben fontos döntéseket hozni. Az üzleti intelligencia és döntések specializáció segítségével megismerkedtem az adatbázisok felépítésével, illetve azzal, hogy hogyan kezeljük őket. A tanórák keretein belül több adatbázist is volt lehetőségem megtekinteni és képes voltam azt szerkeszteni kedvem szerint. Egy fiktív vállalatnak fogok teljesen az alapoktól kezdve felépíteni egy adatbázis modellt, ezzel megmutatva azt, hogy mire is szükséges odafigyelni adathalmazok létrehozásakor, milyen kapcsolatban állnak egymással bizonyos táblák és hogyan kötődnek azok egymáshoz. Úgy gondolom, hogy egy jól felépített adatbázis nagyban meg tudja határozni egy cég működését.

Egy vállalat működése nagyban meghatározza azt, hogy milyen adatbázis modellt fog alkalmazni, de ehhez kell egy megfelelő adatbázis kezelő szoftver is, ami kezeli magát a rendszert. Érdeemes megvizsgálni azt, hogy a multinacionális cégek mely adatbázis kezelőket választják és milyen szempontok alapján, hiszen így mi is képesek leszünk eldönteni, hogy nekünk is megfelel-e az

általuk használt alkalmazás és struktúra. Elengedhetetlen egy ilyen rendszer létrehozásakor az adatbiztonsági kérdések megválaszolása és annak tisztázása, mert értékes adatokat tárolnak a szervezetek és ezek nem megfelelő biztosítása súlyos következményekkel járhatnak.

A diplomamunkámban bemutatom azokat a gyakorlatokat, amiket cégek alkalmaztak az elmúlt években és az új trendeket, amikkel tudjuk a rendszerek teljesítményét és hatékonyságát növelni és ez milyen hatással lehet a felhasználókra, megvannak-e velük elégedve.

Hipotézisem az, hogy a multinacionális és nagy vállalatok többsége a relációs adatbázisokat alkalmazza az adatai tárolásához és kezeléséhez, emellett számukra fontosabb a teljesítmény, mint a költséghatékonyság. Fontos arra is kitérni, hogy van-e hajlandóság a felhő alapú adatbázisokra való áttérésre vagy pedig a megszokott székhelyen lévő szervereket érvényesítik előnyben. Mindemellett bemutatom az adatbázisok múltját és azok fejlődését is.

Kutatásom a szekunder kutatási módszeren és egy mély interjún fog alapulni, mert meg kell ismernem egy vállalkozás működését ahhoz, hogy képes legyek annak előállítani egy adatbázist. Az interjút egy tapasztalt személlyel készítettem el, aki saját vállalkozást működtet a mai napig, ami egy kisboltból indult, de mára már egy logisztikai raktárra nőtte ki magát.

Dolgozatom egy fiktív vállalat bemutatásával fog kezdődni: mivel foglalkoznak, milyen táblákra van igényük és milyen adatokat szeretnének a rendszerükben tárolni. Az adatbázis felépítése során fontosnak találom azt, hogy a logikai háttérrel is bemutassam az olvasó számára, hiszen egy általam elképzelt rendszer működését kell megértenie. Beszélni fogok alapfogalmakról, az adatbázisok felépítéséről, érinteni fogom a mesterséges intelligencia használatát és annak hatásait a vállalatokra. Bemutatok egy adatbázis kezelő rendszert, illetve az SQL programozási nyelvet és annak alkalmazását. Végezetül a kész terméket összegezném, bemutatnám a működését, valamint átnyújtanám azt a fiktív vállalatnak.

1. A FIKTÍV VÁLLALAT

1.1. Információ gyűjtés és kutatás

Ahhoz, hogy ténylegesen tükrözni tudjak egy vállalatot, meg kell ismernem a szokásaikat, illetve meg kell terveznem mire is lesz szükségem ahhoz, hogy egy hasonló adatbázist tudjak felépíteni. A modellezéshez tapasztalatszerzés céljából kérdeztem a „Bótimacska ABC” egyik tulajdonosát, Váradi Attilát. A Bótimacska ABC egy kis élelmiszerbolt volt, ami 2022. őszén már bezárt. Attilát arra kértem, hogy általánosságban próbáljon meg nekem beszélni a boltjuk raktár, illetve egyéb készleteiről, hogy ezzel képes legyek az adatbázisomat adatokkal és rekordokkal feltölteni.

Az interjú keretein belül az egykori cégvezetőt a kutatásom szempontjából fontos vásárlói szokásokról is kérdeztem, vagyis, hogy a vevők milyen termékeket, milyen gyakorisággal és milyen mennyiségben vásároltak, hiszen ezek a komponensek lényegi elemeit képezik az elkészítendő adatbázisnak. Általában különböző tényezők befolyásolhatják a vásárlói szokásokat, mint például az, hogy milyen évszak, ünnep vagy időjárás van az adott napon. Nyáron több víz szokott fogyni, abból is inkább a nagyobb 1,5 literes kiszereles. Üdítőket és gyümölcsleveket mindig vesznek a vásárlók, azonban ünnepekkor (karácsony, húsvét) ez a mennyiség megnő. Olyan árucikkekből, mint a wc papír és a papírtörölő, sokkal többet kell tartani raktáron, hiszen ezeket rendszeresen fogyasztják a vevők. Attilának elgondolása a különböző termékcsoportokról változatos volt, például befőttből és savanyúságból két eltérő árfekvésű terméket – egy olcsóbb és egy drágább fajtát – tartottak, hogy megtalálják ezzel a célközönséget. Édességekből nem raktároztak, viszont karton mennyiségben vettek belőle, hogy a készlet ne tudjon gyorsan elfogyni. A tisztítószer is hasonlóan voltak jelen a boltban, raktárkészlet nem volt belőlük, csak az, ami a vevőtérben volt megtalálható. Mivel a cégnek nem volt alkalmazottja, az adatbázisban csak a két tulajdonos nevét kellett tárolni. Rendszeresen ugyan avval a beszállítóval (vendőrral) dolgoztak, heti vagy kétheti rendszerességgel érkeztek az üzlethez, hogy a megrendelt árukat leszállítsák.

1.2. Miért van szüksége egy vállalatnak egy adatbázisra?

A cégek életében fontos szerepet játszik egy jól felépített adatbázis, mely lehet egy komolyabb vállalatirányítási rendszer része, de lehetséges az is, hogy csak önmagában használjuk azt.

Az adatkezelés az egyik alapköve az adatbázisnak és szerepe nélkülözhetetlen. Az adatbázisok egy olyan környezetet biztosítanak, ahol az adatok strukturáltan vannak elhelyezve és képesek vagyunk azokat rendezni és kezelni. Ebbe bele tartozik a vásárlókról tárolt és termékekről szóló

információ, a gazdasági adatok, a raktárkészlet, a gyártási folyamatok ellenőrzésének eredményei, a dolgozói adatok és még hosszasan lehetne sorolni. Ha megfelelően van kezelve az adat és az adatbázis, akkor képesek vagyunk vele kulcsfontosságú döntéseket hozni és bizonyos folyamatokat optimalizálni.

Az adatintegritás és konzisztencia az adatbázison belül szabályoznak egyedi (unique) és idegen (foreign key) kulcsokkal. Az adatintegritás fenntartása jelenti a megbízható üzleti háttérrel ahhoz, hogy folyamatokat és riportokat tudjunk készíteni.

Az adatbiztonság üzleti szempontból elengedhetetlen. Rengeteg érzékeny információval rendelkezik egy adatbázis, például a dolgozók személyes adatai vagy pedig bérezésük. Az adatbázisok különböző funkciókkal vannak ellátva, hogy az adatokat biztosítsák; ezek lehetnek például a felhasználói belépési kérelmek, a dolgozói hozzáférések szabályozása vagy pedig az adatok titkosítása. Nem csak az érzékeny adatoknak, hanem az egész adatbázisnak szüksége van arra, hogy biztosítva legyen. Minden ügyfélnek fontos az, hogy a saját adatai ne kerülhessenek harmadik félhez (GDPR), mert ez bizalom veszteséssel járhat.

A vállalkozások zavartalan működése szempontjából szükséges felkészülni a legrosszabb eshetőségre: az adatvesztésre. Az adatbázisokban létezik biztonsági mentés és annak helyreállítási funkciója, ami arra szolgál, hogy ha a rendszerben hiba keletkezne, képes legyen egy korábbi állapot visszaállítására. Ezek a mentések fontos szerepet játszanak abban, hogy azt az időt minimalizálják, amíg a rendszer nem volt működőképes és a lehető leggyorsabban tudják visszaállítani az utolsó mentett állapotot.

Az adatelemzés gazdasági szempontból is sok segítséget nyújthat a vállalat életében. Az adatbázis szolgál az üzleti intelligencia (business intelligence) és elemzések alapjául. Mivel az adatok strukturált formában vannak tárolva a cégek képesek komplex riportokat vagy lekérdezéseket generálni, ezzel fontos stratégiai döntéseket tudnak hozni.

A vállalkozások növekedésével megnő az adatmennyiségük is. Az adatbázis lehetőséget nyújt felhasználó részére a nagy adathalmazok kezeléséhez és a megnövekedett terhelés befogadásához. Sok esetben az adatbázisok lekérdezési ideje növekedni kezd, ha sok oszlopot használunk egy táblán belül vagy minden információt egy helyre sűrítünk. Természetesen ez a sebesség függ attól is, hogy mekkora adatmennyiséggel kell dolgoznunk, így érdemes a már nem aktuális adatokat archiválni, ezzel tudjuk a rendszer teljesítményét növelni vagy szinten tartani azt.

1.3. Egy fiktív vállalat felépítése

Az Iguana.pub KFT. legyen egy kiskereskedelmi vállalat, amely élelmiszerek és vegyiárak vételével, valamint eladásával foglalkozik. A bolt egy 20.000 fős kis város szélén helyezkedik el, ahol naponta nem jelenik meg túlzottan sok vásárló, azonban, hogy versenyképes tudjon maradni a környéken jelen lévő multinacionális cégek mellett, vállalja, hogy ünnepnapokon is nyitva tart.

Az üzlet tíz dolgozót alkalmaz, akik reggeli, napközbeni és esti nyolcórás műszakokba vannak beosztva, egy óras munkaszünettel. Általában egyszerre három-öt alkalmazott tartózkodik a munkavégzés helyén a forgalmi igényeknek megfelelően. Családbarát munkahely lévén figyelembe veszik a kisgyermekes szülők olyan kéréseit is, melyek a túl korai, vagy a túl kései műszakban való tartózkodásra vonatkoznak.

A vállalat több különböző beszállítóval áll kapcsolatban, akik eltérő árakon biztosítanak számos terméktípust és terméket a kft. számára. Így fordulhat az elő, hogy időközönként megváltozik melyik vendor-tól szerzi be a cég az adott árut, evvel törekedve a profitmaximalizálásra. A szállítóval megtörténik az elszámolás, ha szükséges a termékekért és a fuvarért fizetni azt helyben rendezik, kivéve akkor, ha az már előre utalással már megtörtént. A beérkezett árut regisztrálják az adatbázisba.

Minden eladásra kerülő terméket a rendszer automatikusan eltávolít a raktárkészletből, így nem kell manuálisan elvégezni ezt a műveletet. Azt is követi az alkalmazás, hogy ki volt az, aki a terméket eladta, mennyit adott el belőle, készpénzes vagy kártyás fizetés volt-e, esetleg megosztotta és mindkét szolgáltatást igénybe vette-e. Az eladott termékek után a dolgozók premizálhatók.

A raktárkészletben számontartják azt, hogy a termék szavatossági idejét, hány darab található belőle az üzletben, mekkora kiszerelesről van szó (0.5 liter vagy pedig 1.5 liter, 0,250 dkg vagy pedig 1kg), az aktuális eladási és beszerzési árát, valamint típusát (gyümölcs, zöldség, folyadék, édesség).

Biztonsági mentéseket nem készítenek az adatbázisukról, mivel még a papír alapú rögzítést sem vetették el és szimultán használják a kettőt. Úgy gondolják, hogy ez így sokkal jobban ellenőrizhető.

A versenyképesség fenntartása és az árbevétel optimalizálása érdekében riportokat készítenek a termékadásokról a konkurencia és a fizetőképes vásárlók nyomon követésével. A termékek árait ezek figyelembevételével is kívánják alakítani a jövőben.

A fiktív vállalatnak egy, a szükségleteit kielégítő adatbázisra lesz szüksége, ahol képesek tárolni a dolgozók, termékek és a beszállítók adatait. Tudják követni rendelések esetén, hogy mikor rendelték az árut, az mikor érkezett be, illetve, hogy van-e fennálló tartozásuk a szállító felé az áru beérkezésekor. Tudják mérni azt is, hogy a boltban dolgozók, mikor érkeznek és mennek el, valamint, hogy melyik műszakba voltak beosztva.

2. ADATBÁZISOK FELÉPÍTÉSE

2.1. Az adat és az adatbázis

Mielőtt elkészítenénk egy fiktív vállalat adatbázisát, tekintsük át mit is kell tudnunk először magáról az adatról és az adatbázisok felépítéséről.

„Azokat az információkat, amelyekből valamilyen konkrét tényt tudunk meg adatnak is nevezzük. Az információ értelmezett adat. Az adat elemi ismeret. Az adat tények, fogalmak olyan megjelenési formája, amely alkalmas emberi eszközökkel történő értelmezésre, feldolgozásra, továbbításra. Az adatokból gondolkodás vagy gépi feldolgozás útján információkat, azaz új ismereteket nyerünk.” (Gál, dátum nélk.)

Az adatoknak különböző típusaik lehetnek és ezek csoportokba rendezhetőek a felhasználási céljaik szerint, például a bennük használt karakterek típusa alapján:

- alfabetikus
- alfanumerikus
- numerikus

Ezen felül lehetséges még adatokat folyamatra gyakorolt hatása (mennyiségi, azonosító vagy leíró jellegű), ábrázolás és rögzíthetőség (digitális vagy analóg), illeszkedés az adatfeldolgozás során (formázott vagy formázatlan adatok), folyamatban elfoglalt helyük (bemeneti vagy kimeneti), folyamatbeli szerepük (törzs, készlet és mozgásadatok), programok által értelmezett típusok (egész és valós számok illetve, szöveg, dátum, pénznem, feljegyzés, számláló, OLE objektum, hipervivatkozás, igen/nem rekordok) alapján is megkülönböztetni. (Dr. Budai, 2020)

Amennyiben több adatról beszélünk azt adatállománynak nevezzük, abban az esetben, ha azok egymással kapcsolatban állnak vagy pedig egységesnek tekinthetőek valamely szempontrendszer alapján. Ezeket az adatállományokat pedig szeretnénk lehetőleg egy helyen tárolni, az adatbázis pedig pontosan erre a célra szolgál, segítségével ezeket az adatállományokat, ha kölcsönösen összefüggésben állnak egymással, képesek vagyunk egy strukturált rendszerbe helyezni. (Fogalomtár, 2014) Adatstruktúrák lehetnek rekordok és tömbök. Rekordnak hívunk különböző típusú adathalmazokat, melyben az egyed összes tulajdonságát megtalálhatjuk egy adatbázishoz tartozó táblázat egy adott sorában. Tömbnek olyan adatállományt nevezünk, aminek elemei

azonos típusú (például numerikus, és ilyenek lehetnek a mátrixok vagy vektorok) adatokat tartalmaznak. (Dr. Budai, 2020)

Az adatok kezelését kétféleképpen tudjuk csoportosítani az ismeretkezelés alapján, adatszerű vagy szövegszerű. Az adatszerű ismeretkezelés egy adatbázis állományaiban történik, ahol a rekordok (táblázat sorai) vagy mezők (táblázat egy-egy cellája) között végzünk műveleteket. Szövegszerű ismeretkezelés esetén tezaurusz (specifikus területen használatos szakkifejezések, vagy másnéven szókincstár), vagy deszkriptor (definíciók leírására alkalmas ismeretrészeket leíró rész) módszerek közül választhatunk. (Dr. Budai, 2020)

Az adatbázis egy „számítógépen tárolt, egymással bonyolult összefüggésben álló adatok rendszere, összetett logikai adatszerkezet fizikai leképzése, amely tartalmazza az adat- és szerkezetleírást, valamint az adatkezelő eljárásokat is”. (Dr. Budai, 2020) Egy adatbázis két részből tevődik össze: egy adathalmazból és egy hozzá tartozó adatbáziskezelő szoftverből (az én esetemben ez az SQL Server). A program feladata az, hogy az adatbázis tartalmát megismételhetően megtalálja és annak adatait ki tudja olvasni, meg tudja változtatni, vagy esetleg újat képes létrehozni, beszúrni. Az adatbázisok önálló nyelve az SQL – Structured Query Language, ami magyarra fordítva Strukturált Lekérdezési Nyelvet jelent. Bár az SQL, mint szabvány nyelvezet van jelen, az adatbázis kezelő rendszerek saját dialektus implementációkkal rendelkeznek, például MySQL, Oracle, PostgreSQL és MSSQL. Ezek az egyéni dialektusok kisebb vagy nagyobb eltéréseket mutathatnak a standard nyelvhez képest. Az adatbázisokat képesek vagyunk modellezni hierarchikus, hálós, relációs és objektum orientált módokon. (Dr. Budai, 2020)

2.2. A relációs adatbázis modell

A relációs adatbázis modell lesz számomra a legrelevánsabb a kutatás során, hiszen bizonyos termékek vagy eladók között lesz kapcsolati viszony. A kapcsolatok az alábbi tulajdonságokkal vannak felruházva:

- Egyértelmű névvel rendelkezik
- Minden sorban azonos mennyiségű oszlop van
- Minden oszlopnak egyértelmű neve van
- Az oszlopok sorrendje nincs hatással az adatmodellre
- A táblázatnak nem lehet kettő megegyező sora
- A sorok száma meghatározza a reláció méretét

- Minden rekord minden oszlopa legalább alapértelmezéssel bír
- Minden reláció rendelkezik egy olyan oszloppal, amely meghatározza a sor további oszlopainak az értékét

A relációs adatmodellben az azonosítóra bizonyos feltételek vonatkoznak:

- Minden egyednek rendelkeznie kell azonosítóval
- Az azonosító értéke nem lehet üres vagy ismeretlen (NULL érték)
- Minden egyednek kizárólag egy azonosító tulajdonsága lehet
- Egy tulajdonság csak egy egyednek lehet az azonosítója

Ezen szabályok alapján megtudjuk határozni, hogy számunkra mi a megfelelő elsődleges kulcs. Egyedi, az egyedre erősen jellemző, stabil, struktúrája folyamatos, rövid, hosszúsága fix és a benne szereplő karakterek intervalluma és számossága is szűkös. Ezért minden táblában egy egyedi ID mezőt alkalmaztam erre a célra, amely egész szám típusú (int). (Dr. Budai, 2020)

A relációs adatbázisoknak a jellemzői közé tartoznak olyan tételek is, amiknek az értéke az egyed előfordulására tekintve nem egyedi, így ők csak leíró szereppel bírnak. Olyan eset is van, amikor az egyed egyik tulajdonságának az egyik táblában azonosító, míg egy másik táblában leíró szerepe van. Ezt a tulajdonságot a leíró feladat betöltése esetében, kapcsoló szerepű tulajdonságnak nevezzük. (Dr. Budai, 2020)

Ha az egyed rendelkezik jellemzésében és kezelésében kitüntetett szerepet betöltő tulajdonsággal, akkor azt kulcsnak nevezzük. Az egyedet többféleképpen írhatjuk le és kezelhetjük, így egy adott egyednek több kulcsa is lehet. Ezeket kategorizálni is tudjuk. A kulcs, mint fogalom csak a típusnevével együtt értelmezhető: azonosító, kapcsoló, rendezési, csoportosító és szelekciós kulcsokat ismerünk. Ezek közül a leggyakrabban használt típus az azonosító és kapcsoló típusú kulcsok. (Dr. Budai, 2020)

Relációkban előfordulhatnak bizonyos anomáliák, amiknek a megoldása az, hogy normalizálással a kívánt formára hozzuk a relációkat. Ilyen fajta anomália lehet: beírási (adatismétlés, redundancia), módosítási (új rekord felvételekor keletkező probléma), valamint a törlési (ahogy a neve is mondja, információt törölünk az adatbázisból, például adatkarbantartás során). (Dr. Budai, 2020)

A kulcsokkal képesek vagyunk kapcsolatokat létrehozni, amiknek pedig vannak irányaik. Ezek a kapcsolattípusok adatbázisonként és táblánként is változhatnak. Léteznek egyik irányban egyértelmű, másik irányban többértelmű (1:N, például tulajdonos - gépkocsi), több és többértelmű (M:N, például tulajdonos - kocsitípus), kölcsönösen egyértelmű kapcsolat (1:1), valamint van olyan eset is, amikor a két elem egymástól teljesen független. (Dr. Budai, 2020)

Relációs adatbázisunk tervezése során két módszert is tudunk alkalmazni: ezek az analitikus (ER modell) vagy szintetikus (SGM) módszerek. Én az utóbbival fogok dolgozni, hiszen a tanórák során ezzel ismerkedtem meg és ezzel dolgoztam leginkább. (Dr. Budai, 2020)

Az adatbázisunkhoz olyan többfelhasználós környezetet kell választani, ami figyelembe veszi az ACID elveket, és egy biztos alapot tud nyújtani bármelyik adatbázis modellnek. Az ACID egy mozaik szó, ami a következőt jelenti: atomicity, consistency, isolation és durability. Ezeknek fontossága elengedhetetlen. Az atomicity (atomikusság), amennyiben hibás tranzakció meggyégbe, visszaáll az előző állapotába. A consistency (konzisztencia), miután befejeződött a tranzakció az adatbázisnak muszáj konzisztens állapotba kerülnie. Az isolation (elszigeteltség), a különböző párhuzamosan működő tranzakcióknak egymástól függetlenül is kell tudniuk funkcionálni. A durability (állandóság, tartósság), a már lezárt tranzakcióknak az eredményei nem veszhetnek el. (Dr. Budai, 2020)

2.3. További adatbázis modellek

A hierarchikus adatbázis modell egy úgynevezett „fastruktúrikus” formában tárolja az adatait, amelyek alá és fölérendeltségi, másnéven szülő - gyermek viszonyban állnak egymással. Ezt a fajta struktúrát az IBM fejlesztette ki a 60-as években (még az adatbázis kezelő rendszerek korai fázisában) és a már korábban említett 1:N-hez elvet alkalmazza. A szisztéma felülről a gyökérből (kiindulási pont) indul lefelé, ahol minden adat rendelkezik kizárólag egy szülővel és tovább haladva nulla vagy attól több gyermeke lehet. A hierarchikus adatbázisok előnye, hogy hatékonyak lehetnek a szülő - gyermek viszonyok kezelésében, ami különösen hasznos lehet, például természeti vagy szervezeti hierarchiák reprezentálásában. Azonban hátrányt jelenthet a többi adatbázis modellel szemben, hogy nehezen kezelik a változó adathalmazokat és a bonyolultabb lekérdezéseket. Ez azt jelenti, hogy az adatok nem feltétlenül tudnak közvetlen kapcsolatban állni egymással és adattörlés esetén „elárvulhat” egy adattömb. Ilyen esetben a leválasztott adatok nem tudnak kommunikálni a „fa” többi részével. (Hillyer, 2011)

A hálós adatbázis a 70-es években jelent meg és a hierarchikus modellhez képest, már sokkal rugalmasabbnak bizonyult abból a szempontból, hogy a különböző adatelemek már képesek egymással közvetlen kapcsolatban is állni. Ez a korreláció az adatok között egy háló formájában jelenik meg és reprezentálja az M:N-hez viszonyt. Az adatbázisban lévő rekordoknak saját belső felépítéssel rendelkeznek, ami kettő vagy több általános adatot kategorizál, majd hoz létre egy adatcsoportot. Ezek a csoportok nem csak az általános adatokból állnak, hanem más csoportokat is bevonnak azért, hogy kiépítsenek egy tábla struktúrát. A tábla a csoportosított értékek sokasága egy adatelem neve alatt, ami lehet például termék, dolgozó vagy akár beszállító is, attól függően, hogy milyen adatbázisról beszélünk. Előnyei ellenére, hátrányt jelenthetnek a bonyolultabb lekérdezések és a struktúra karbantartása, mivel, ha egy elemet elveszünk a rendszerből, akkor előfordulhat, hogy az az ACID elveknek fog ellentmondani. (Maurer & Scherbakov, 2006)

Az objektumorientált adatbázis modell az objektumelvű programozási nyelv megközelítéseit alkalmazza az adatok kezelésére és szervezésére. Az előző adatbázisokban megszokottakhoz képest, a táblák kicserélődnek a korábbiaknál sokkal komplexebb adatokra, amelyeket objektumoknak nevezünk. Ilyen objektumok lehetnek képek, hangok, videó anyagok és akár összetettebb struktúrák is, amiket az adatbázis már képes gond nélkül tárolni. A modell sajátossága, hogy az adatokat osztályokba és objektumokba szervezi, amelyek tulajdonságokat (mezők/attribútumok) és műveleteket (metódusok) tartalmaznak. A táblákhoz hasonlóan, az objektumok is állhatnak egymással kapcsolatban és helyezkedhetnek el komplikált struktúrákban. Az objektumorientált adatbázis modell használati előnyei közé sorolható az adatok összetettségének és az alkalmazások logikájának jobb tükrözése, valamint a fejlesztési idő csökkenése. Az adatmanipulációs eszközök és az adatok együttes kezelésével viszont (nem muszáj külön-külön műveleteket futtatnunk), sokkal jobban megterhelhető a rendszer és a bonyolult struktúra itt akár vissza is üthet. (Garcia-Molina, D. Ullman, & Widom, 2008)

A NoSQL (Not Only SQL) adatbázisrendszerek eltérnek az előzőekhez képest az adatok tárolása és azok kezelésének szempontjából. Ezt a típusú adatbázist leginkább a rendszer skálázhatósága, a teljesítménye és a rugalmassága jellemzi, mely leginkább nagy mennyiségű adatok kezelésekor és a változó adatsémákhoz való alkalmazkodáskor mutatkozik meg. A NoSQL-en belül több főbb típust is fel lehet sorolni: kulcs-érték tárolók; dokumentum-alapú adatbázisok; osztott oszlopos adatbázisok; és gráf adatbázisok. Egyszerű adatmodell, ahol az adatokat kulcs-érték párban kerülnek tárolásra. A dokumentum-alapú adatbázis használatával képesek vagyunk már JSON

vagy XML dokumentumokat is tárolni a modellben, ugyanis az alkalmazások könnyedén tudják ezeket a fájl formátumokat kezelni. Az adatokat osztott oszlopokban is tárolhatja a rendszer, mely hatékonyan bizonyul a lekérdezésekhez és a nagy adathalmazokhoz való skálázódáshoz. A gráf szerkezet különösen alkalmas az összetett kapcsolatok és hálózatok lekérdezésére. Számos olyan előnye van a NoSQL adatbázisnak, mint például a rugalmasság, a skálázhatóság és a gyors fejlesztési ciklusok, ellenben fontos megérteni az alkalmazás specifikus igényeket és a rendszer adta lehetőségeket, hogy a számunkra legmegfelelőbb technológiát tudjuk választani. (J. Sadalage & Fowler, 2013)

2.4. Az SGM, valamint annak jelölései és szabályai

A szintetikus tervezési módszer (másnéven SGM) lehetőséget biztosít nekünk tulajdonságok és belső függések felvételére, normalizálásra, adatszerkezeti ábrák szerkesztésére és ellenőrzésre.

Az SGM szerkezete nagyon egyszerű és az alábbi módon épül fel (1. ábra):

- Sorszám
- Tulajdonság neve (tartomány leírás)
- Hivatkozás
- Típus, hossz
- Rövid név (mezőnév)

SGM Tábla

Tulajdonság név		rövid név	01	02	03	04	05	06	07	08	09	10
1	ügyfél címe	UCIM	←									
2	ügyfél kódja	UKOD	*			←						
3	ügyfél neve	UNEV	←									
4	bankfiók címe	BACIM		←								
5	bankfiók kódja	BAKOD		*		←						
6	szerződés száma (azonosítója)	SZESZ				●	●	●				
7	szerződés típuskódja	SZETIP			*	●	●	●				
8	szerződéses összeg	SZOSSZ				←						
9	szerződés dátuma	SZEDAT				←						
10	szerződéstípus megnevezése	SZETIM			←							
11	törlesztés esedékessége	TORESZ						←				
12	törlesztés összege	TOSSZ						←				
13	törlesztés sorszáma	TOSOR						●				
14	folyósítás esedékessége	FESED					←					
15	folyósítás összege	FOSSZ					←					
16	folyósítás sorszáma	FOLYOS					●					
Jelölések: * egyszerű azonosító ● részazonosító ← leíró tulajdonság (funkcionális függés) ↔ azonosító jelölt (kölcsönös függés) < gyengén jellemző tulajdonság			TÁBLA NEVE ÜGYFÉL BANKFIÓK SZERZŐDÉSTÍPUS SZERZŐDÉS FOLYÓSÍTÁS TÖRLESZTÉS									

Forrás: Dr. Budai László, 2020

A táblán belül tudunk meghatározni egyszerű (*) (korábban már említett azonosítóként fogom hívni) és- összetett azonosítókat (●) (más néven részazonosító, ebből kettőt vagy többet is el tudunk helyezni egy adott sorban), illetve leírotulajdonságokat (←), melyeket a leggyakrabban használjuk az SGM táblák készítésekor. Ezeknek különböző jelölésük van és mindegyikhez bizonyos szabályok tartoznak, amiket be kell tartanunk. Fontos, hogy egy oszlopban nem állhat egyszerre azonosító és részazonosító. Minden sorban kötelező szerepelnie legalább az egyik fentebb említett eszköznek, hogy bele tudjon tartozni magába az adatbázisba. Ha nem lenne jelölése az adott sorban, akkor nincsenek összefüggésben az adatok. Az oszlopok 3NF relációt deklarálnak. Ha a sorokban több jelölés van, legalább az egyiknek azonosítónak kell lennie, mivel így tudnak egymáshoz kapcsolódni a táblák, ha nem lenne egyik sem egyszerű vagy összetett azonosító, akkor nem tudnánk beazonosítani a más táblák béli rekordokat vagy adatokat. Az egy sorban lévő jelek a határozzák meg a relációk közötti kapcsolatot. Közvetlen kapcsolat csak és kizárólag az azonosítóval ellátott oszlop rendelkezik, illetve azok az egyedek és oszlopok, amik ennek megfelelően vannak jelezve. Ez a kapcsolódás csak egy azonosító és egy nem azonosító (például leíró tulajdonság) jel között jöhet létre. Amennyiben több összetett azonosítóról van szó, azokat egy sorként kell kezelni. (Dr. Budai, 2020)

2.5. Az adatbázisok normalizációja (NF)

Az adatbázisok normalizációja a relációs adatbázisok tervezéséhez tartozik, ami az adatok redundanciájának és inkonzisztenciájának (ACID elv) minimalizálásáért felel. A rendszer normalizálása azt jelenti, hogy az adatok táblákba szerveződnek, hogy ezáltal minimalizálják a redundáns információkat és javítsák az adatbázis hatékonyságát. A redundancia alatt azt értjük, hogy egy adott információ többször is előfordul egy adatbázison belül a sorokban, az oszlopokban vagy akár a táblarendszerben. Sor redundancia az, amikor egy táblában többször tároljuk ugyanazt az adatot (például ügyfél adatait több helyen tárolják, ezzel ismétlődik a neve és a címe is). Oszlop redundancia esetén több oszlopban is előfordul ugyanazon információ. A táblarendszer pedig arra vonatkozik, hogy több táblában is tárolásra kerül az adat (például két különböző tábla ugyanazokkal a tulajdonságokkal rendelkezik és egyforma információt hordoznak magukban ugyanarról a személyről). A redundancia megnöveli az adatbázis méretét, ezáltal több helyet foglal el és veszélyt jelent a konzisztenciára is; ezért a tervezés során érdemes inkább a minimalizálásra törekedni. Ebben nyújt nagy segítséget a normalizáció, aminek három formáját mutatom be. Az első (1NF), normál formában történő normalizáció minden attribútum atomi értékek halmaza, mely azt jelenti, hogy egy mezőben nem jelennek meg ismétlődően csoportok vagy többértékű tulajdonságok. A 2NF-ről akkor beszélünk, ha minden tulajdonság függ a tábla elsődleges kulcsától. A harmadik (3NF) formában pedig nincsenek tranzitív függőségek az egyszerű azonosító és a leíró tulajdonságok között, melyek nem tartoznak közvetlenül az elsődleges kulcshoz (ebben az esetben az adat egy másik táblához van rendelve). Az normalizáció és az adatmodellezési elvek betartása fontos, hiszen ez a terv segíthet minimalizálni az adatbázisok redundanciáját és javítani a konzisztenciát, ezzel is hozzájárulva a hatékony adatmanipuláció és adatintegritás fenntartásához. (Kent, 1982)

3. ÚJÍTÁSOK ÉS TRENDEK, MI VÁR RÁNK A JÖVŐBEN

3.1. Technológiai fejlődés az adatbázisok területén

Az adatbázis kezelő rendszerek fejlődése több irányba halad: felhőalapú, kiterjesztett és önműködő irányba. A három közül mára a felhő támogatta rendszerek használata lett a legnyilvánvalóbb, ugyanis széles körben elterjedt (ilyen például a Microsoft Azure felülete).

Több vállalkozás használ különböző adatbázis rendszereket, mint például relációs adatbázisokat, adattárházakat és NoSQL-t. Ennek következtében a cégeknél lévő adminisztrátorok több rendszert kell, hogy felügyeljenek a felhő szolgáltatásokkal szimultán. (Foote, 2023)

Az adatbázisokat kezelő rendszerek elsősorban szervezetek adatainak mentésére és azok strukturált rendszerezésére szolgálnak. Az 1960-as évekbeli kezdeti, egyszerű adatbázisok hierarchikusan (egy a sokhoz kapcsolódás) vagy pedig hálózati (több kapcsolatot engedélyező) módon épültek fel. (Foote, 2023)

A relációs adatbázisok csak a 80-as években terjedtek el és a 90-es években követték őket az objektumorientált adatbázisok. A ma ismert felhő adatbázisok új lehetőségeket biztosítanak a felhasználóknak és vállalatoknak, hogy adatokat gyűjtsenek, tárolhassanak és azokat kezeljék. Az adatbázis kezelés üzleti szempontból rendkívül fontossá vált. Nagy mennyiségű adatok hatékony tárolása és karbantartása a cél, ennek a haszna pedig kulcsfontosságú. Lényeges, hogy az elemzők aktuális és megbízható adatokból tudjanak dolgozni. Különböző adatbázisoknak más és más olyan kiemelkedő tulajdonságuk van, amik meghatározzák azt, hogy mely munkafolyamatokhoz tudják őket használni. A kor előrehaladtával és az internet fejlődésével az adatbázisok és az adatbázis-kezelők használatának száma is növekszik. (Foote, 2023)

3.2. Új trendek és lehetőségek

Az idő múlásával és a technológia vágató fejlődésének köszönhetően, újabb és újabb eszközöket kapunk ahhoz, hogy a már létező rendszereket képesek legyünk egyszerűsíteni és gyorsítani. Az érték optimalizálás is hasonló módon segíti a munkát. Az értékoptimalizációhoz egy cég adat, analitikai és mesterséges intelligencia portfóliója szükséges és annak integrált értékmenedzsment készlete. (Choubey, 2023)

Az optimalizáció mellett fontos az, hogy az adatraktárak képesek legyenek fenntarthatóak maradni. Ezt pontosan úgy tudják elérni az adatelemzők, ha a saját folyamataikat optimalizálják.

A mesterséges intelligenciák használói tudatában vannak avval, hogy az energialábnymuk rohamosan növekszik, ezért elkezdtek bizonyos gyakorlatokat és szokásokat kialakítani, például a megújuló energia egy remek eszköz arra, hogy ezt csökkentsék vagy pedig szinten tudják tartani. Ehhez csatlakozik még az energiatakarékos eszközök használata, illetve a kisebb adatok és gépi tanulás tudatos bevezetése. (Choubey, 2023)

A mesterséges intelligencia jelenléte hívogató a nagy vállalatok számára, hiszen a számításokat képes egymaga is elvégezni. Természetesen fontos tisztában lenni a velejáró kockázatokkal és azok kezelésével. A ChatGPT és a hozzá hasonló MI alkalmazások olyan új veszélyeket jelentenek a cégekre nézve, mint például az etikai kockázatok (GDPR), a gépi betanításra szolgáló adatok megfertőzése vagy pedig a csalások észlelésének elkerülése. A kockázatkezelés nem csak a szabályozásban merül ki, hanem szükséges az érdekelt felek közötti bizalom kiépítése, amit a hatékony és felelős gyakorlatok fognak megerősíteni, illetve a mesterséges intelligenciát használó személyek továbbképzése ezen a téren hozzájárul a problémák elkerüléséhez. (Choubey, 2023)

Nap mint nap találkozunk adatokkal, amik velünk is meg vannak osztva, legyen az egy kép a közösségi médián vagy pedig a munkahelyen a jövő évi költségvetés keretszámai. Az adatmegosztás manapság már nélkülözhetetlen. Ez magába foglalja a részlegek vagy vállalatok között történő adatáramlást. Szervezetek gyakran hoznak létre adatokat, mint termékeket. Ez számunkra azt jelenti, hogy az adatelemző eszközöket olyan formátumban készítik el, hogy azokat képesek legyenek átadni vagy megosztani másokkal. (Choubey, 2023)

Az angoltól származó "data fabric" olyan adatkezelési megközelítés, amely lehetővé teszi az adatok összekapcsolását, integrálását és kezelését különböző forrásokból és formátumokból. Ez a megközelítés lehetővé teszi az adatok szabad áramlását az adatközpontok, felhők és más rendszerek között, miközben biztosítja az adatok egységes láthatóságát és hozzáférhetőségét. A data fabric segít a vállalatoknak különböző adathalmazokat és adatforrásokat összekapcsolni, optimalizálni az adatok használatát, és javítani az adatkezelés hatékonyságát. Ennek eredményeképpen, ha folyamatosan elemezzük az adatszöveget képesek vagyunk olyan figyelmeztetéseket és ajánlásokat kapni, ezáltal elősegítve a jobb döntéshozatalt és az üzleti teljesítmény javítását, amit a gépek és mi emberek is feltudunk használni. Lehetősége van az üzleti felhasználónak az adat fogyasztására és a data fabric magabiztosságot ad a kevésbé képzett fejlesztőknek, hogy ők is tudják növelni a tudásukat az integrációs és a modellezési folyamatokban. (Choubey, 2023)

A ChatGPT és generatív MI-k hozzák el az évtized a legnagyobb változásait, nem csak nekünk általános felhasználóknak, hanem a nagyobb vállalatok számára is. Meghatározzák azt, hogy milyen ütemben növekszik az adatmennyiség és annak tárolása, valamint, hogy mi hogyan vagyunk képesek ezekhez alkalmazkodni. A mesterséges intelligencia felhasználásának következő lépcsőfoka a szervezetekbe való beépítés és annak alkalmazása. Most még nem lehet tudni azt, hogy ezeket az eszközöket, hol és milyen formában képesek optimalizálás szempontjából beilleszteni egy cég életébe. (Choubey, 2023)

A jövőbeli elképzelések között szerepel a konvergált adatelemzési ökoszisztéma platformok tervezése és telepítése, hogy képes legyen zökkenőmentesen működni egy adatbázis. A megfelelő kialakításnak köszönhetően ezek a rendszerek modulárisak, alkalmazkodóak és rugalmasabbak, ezáltal dinamikusan fejleszthetőek és növelhetőek is. Ez nagyon fontos az üzleti életben, ha a rendszer igénye növekedni kezdene. (Choubey, 2023)

Fontos, hogy a felhasználói igények és a rendszer fejlődése szinkronban történjen. A változtatások során olyan felületeket hoznak létre a célközönség igényeinek kielégítésére, amelyek dinamikusan beágyazottak és más felhasználók tapasztalatain is alapulnak.

Az automatizációnak meg vannak az előnyei és hátrányai is, de ami elengedhetetlen az az emberi ráhatás magára a rendszerre. Nem feltétlenül kell minden döntésünket mesterséges intelligenciákra bízni és automatizálni, az általa kapott információkat lektorálni kell és nem szabad csak a kapott adatokra hagyatkozni. Az adatelemző csoportok kiemelt feladata az, hogy a döntéseket támogassák különböző eszközökkel. A vállalatok számára szükséges, hogy felismerjék ezt a helyzetet és figyelembe vegyék azt, hogy a gépek által vezérelt szervezetnek nincsen lelkiismerete és nem képes döntést hozni a kitűzött cél elérése érdekében. Általánosságban mondhatjuk azt, hogy fontos összekapcsolni a programok által készített analíziseket az emberi döntéshozatallal. (Choubey, 2023)

Gyerekipőben járó új típusú adatbázisnak minősül az önműködő adatbázis. Ő is felhőalapon működik és gépi tanulást használ ahhoz, hogy automatizálja az adatbázisok biztonsági mentéseit, frissítéseit és egyéb feladatait, amit alapvetően emberek végeztek el korábban. Ezek az autonóm adatbázisok képesek elhárítani a kibertámadásokat olyan adatokon és adatbázisokon, amelyek nincsenek titkosítva, ezzel megelőzve a szerverek leállítását. (Foote, 2023)

Az autonóm adatbázisokat megelőzve, technológiailag előrébb járnak a kiterjesztett adatbáziskezelő rendszerek. Ezek a rendszerek nem csak a gépi tanulás segítségét vonják be a munkába, hanem a mesterséges intelligenciát is felhasználják. Ez jótékony hatást gyakorolhat a törzsadatok kezelésre és az adatintegrációra. Az adatbáziskezelő képes olyan komplex feladatokat végrehajtani, mint az adatminőség ellenőrzése, az adattisztítás, az anomáliák észre vételezése és az adatbányászat. A fő célja az algoritmusoknak az, hogy egyszerűsítse és optimalizálja az adatkezelés folyamatát, továbbá hatékonyabbá és könnyebbé tegye a döntéshozatalokat. (Foote, 2023)

A felhő alapú adatbázisok a legnépszerűbb szolgáltatások és egyben a legjobb megoldás az adatok tárolására és kezelésére. Ez leginkább a vállalkozások számára a kedvező, akik a szolgáltatás adta lehetőségeket ki is tudják használni. A Google Cloud Platform vagy a Microsoft Azure adnak remek lehetőséget arra, hogy megtapasztaljuk az előnyeiket a felhő felületeknek. Az adatbázis bővíthetősége egy rendkívül fontos dolog, hiszen nagy szervezeteknek, rengeteg tárhelyre van szükségük, mivel rohamosan növekszik az adatmennyiségük. Nem csak növelni tudjuk a kapacitását a szervernek, hanem képesek vagyunk csökkenteni azt, így nem feltétlenül kell beruházniuk a vállalatoknak drága adat tároló eszközökből. Ez a fajta szolgáltatás lehetőséget ad arra, hogy a számunkra megfelelő adatbázis modellt használjuk, ami az üzleti modellnek megfelel. Sok szempontból megéri felhő alapú adatbázist használni, hiszen ezeknek az árázása rugalmas és csak felhasználó által használt erőforrásért kell fizetniük. Hozzá tartozik a szolgáltató által kínált karbantartás és frissítések elvégzése, ami képes a munkavállalóknak időt és erőforrást megtakarítani. Amíg a redundancia a relációs adatbázisoknál nem egy ajánlott dolog, addig a felhő szolgáltatásoknál egy rendkívül fontos elem, mert egy adatközpontban tárolják ezeket az adatokat és azok szinkronizációjuk egy elengedhetetlen rész, hogy mindig friss adattal tudjanak dolgozni a felhasználók. Ezt a fajta adatbázist könnyedén tudjuk integrálni egy másik felhőszolgáltatással vagy alkalmazással, ezzel lehetővé téve a cégek számára, az összekapcsolt szolgáltatások előnyeiket és funkcionálisait, így képesek agilisabbak és versenyképesek lenni adatkezelés téren. A nem csak előnyökkel, hanem számos kockázattal is jár egy ilyen befektetés. (What is a cloud database?, 2018)

3.3. Az adat, mint termék és a termék, mint adat

A megelőző fejezetben bemutatott fejlődési lehetőségeket már a nagyobb vállalatok is próbálták és próbálják beilleszteni a mindennapi rutinjukba. Mindezt a Harvard Business Review 2022-es esettanulmánya alapján két nemzetközi cégen keresztül fogom szemléltetni.

Az egyikük egy nagy ázsiai bank, aki a legkevesebb erőbefektetéssel szeretett volna adatokat kivonni a már létező rendszereikből és ezt mind feltölteni egy felhőalapú rendszerbe, egy program segítségével. Három évnyi befektetett munka után lett igazán tiszta a szervezet számára az, hogy csak néhány felhasználó tudja igazán értelmezni a nyers adatokat, amiket ők feltöltöttek és azokat is leginkább ad hoc feladatokhoz és elemzésekhez használták. Ezáltal ez a művelet nem hozott semmilyen eredményt a cég számára. A második alany pedig, egy észak-amerikai bank, ahol sajátkezűleg igyekeztek a rendszert összerakni darabról darabra. Nekik sikerült értéket teremteni abból a szempontból, hogy a fogyasztó szegmentációkat tudták fejleszteni és tudtak hatékonyan veszélyeket elemezni, ellenben a végeredmény egy zavaros személyre szabott csatorna lett és nem lehetett egyszerűen átalakítani. Mindkét vállalatnak az ötlete füstbement még úgyis, hogy különböző technikákat alkalmaztak. Felmerül az, hogy jól közelítették-e meg a céljaikat? (Desai, Fountaine, & Rowshankish, 2022)

Új perspektívából kell megvizsgálni a folyamatokat. Abban az esetben járhatnak ezek a vállalatok sikerrel, ha úgy kezelik az adatállományukat, mintha az egy termék lenne, mivel ilyenkor a cégek igyekeznek maximalizálni az árbevételüket. Ezzel olyan adategyütteseket tudnak létrehozni, ami széleskörű közönséghez elér és sok felhasználó igényét képes kielégíteni. Ilyenek például a saját kezelőfelületek jelentésekhez (dashboard), azok elrendezése tartalom, szín és téma alapján. (Desai, Fountaine, & Rowshankish, 2022)

Ez a megközelítés több előnnyel is rendelkezik. A termékeket évről évre igyekeznek jobbra tenni, ugyanis versenyben akarják tartani a piacionlévő többi termékkel szemben. A produktumokat továbbfejlesztik és egyre több funkcióval látják el őket. Ehhez hozzájárulnak a felhasználók visszajelzései, értékelései, illetve a versenytárcák újításai is. (Desai, Fountaine, & Rowshankish, 2022)

Azok a vállalatok, amelyek az adataikat termékként kezelik csökkenteni tudják az új eseteknél a kutatáshoz felhasznált időt és a költségeiket, ezenfelül csökkenteni tudják a kockázatokat és az adatkezelés által keletkezett terheket. Az adattermék egy olyan eszköz és adathalmaz, amihez szervezetek és felhasználók könnyen elérhetnek és különböző folyamatokhoz képesek

felhasználni. Az adattermékek lehetnek fogyasztók online és bolti vásárlásai, a róluk tárolt adatok (például életkora és neme), hogyan fizetettek és hol. Ezeknek az információknak a birtokában létre tudunk hozni egy klónt, amivel képesek vagyunk méréseket végezni a saját termékünkre vetítve. (Desai, Fountaine, & Rowshankish, 2022)

Számos felhasználási lehetősége van az adatoknak és nagy mennyiségű hozamot képesek generálni. Egy nemzeti banknál egyetlen adat egy vásárlóról képes akár hatvan esetet is lefedni, például valós idejű hitelkockázati felmérés vagy pedig chatbottól való segítségkérés. Ezek felhasználása akár éves szinten 40 millió dollárt is meg tud megspórolni ezeknek a nagy vállalatoknak. (Desai, Fountaine, & Rowshankish, 2022)

Az adattermékek a végfelhasználók számára elképesztően hasznosak, hiszen nem kell nekik időt pazarolni az adatok keresésére, annak megfelelő formátumra való elkészítésére és nem kell létrehozniuk testreszabott adatállományt, mert azt már készen kapják meg. Minden termék arra szolgál, hogy képes legyen a különböző igényeket kielégíteni. Innentől kezdve a fogyasztó és a felhasználó alatt nem embereket, hanem rendszereket értünk, melyeknek több típusa is van. (Desai, Fountaine, & Rowshankish, 2022)

A digitális alkalmazásoknak olyan adatokra van szüksége, amik már megvannak tisztítva (nincsen bennük NULL érték és hibás adat) és olyan formátumban vannak tárolva, ami az adott rendszert kívánja szolgálni. Ilyen lehet például egy tárgy nyomon követése vagy a böngészőkben található sütitik (cookies) is, amik a marketing fejlesztésére vannak kitalálva. Ezek mind olyan adatok, amiknek muszáj up-to-datek lenniük, ezért mindig frissek és frissülnek, megfelelvén a valós időnek. Ahhoz, hogy ezeknek az információknak a birtokába kerülhessünk, elengedhetetlen hozzáférést szereznünk a szükséges alkalmazásokban. (Desai, Fountaine, & Rowshankish, 2022)

Az analitikai rendszereknek is szükségük van a jó minőségű tisztított adatra és arra, hogy azok minél gyakrabban frissüljenek. Ezeket a rendszereket célszerű már úgy megtervezni, hogy egy gépi tanulás vagy pedig egy mesterséges intelligencia képes legyen azt feldolgozni. (Desai, Fountaine, & Rowshankish, 2022)

A jelentési rendszereknél is igaz a már fentebb említett tulajdonság annyival kiegészítve, hogy ezeknek szüksége van az emberi és gépi szabályozásra is. Sok esetben az adatokat csoportosítva szokták átadni, de a vállalatok jobbnak látják azt, ha a modelljeiket napközben frissítik, hogy valós idejű számokat kaphassanak. (Desai, Fountaine, & Rowshankish, 2022)

Felfedező területnek hívják azokat az aggregált adatokat, amiket ad hoc elemzésekre és jelentésekre használnak az adatelemzők és adatmérnökök, hogy ezáltal új lehetőségeket kutassanak és tárjanak fel. (Desai, Fountaine, & Rowshankish, 2022)

A külső adatmegosztó platformoknak nehéz dolguk vannak, hiszen szabályozáson mennek keresztül és rendkívül szigorúak az irányelveik. Különböző megállapodásokat kell betartaniuk arra szándékosan, hogy hogyan tárolják az adatokat, kezelik és védik meg őket. Erre remek példa, hogy a bankok egymással megosztják a csalásokról szóló információikat vagy pedig a kereskedők hogyan fejlesztik a logisztikai rendszereiket. (Desai, Fountaine, & Rowshankish, 2022)

Bizonyos fogyasztási architektúráknak csak és kizárólag egy fajta adat felel meg. Szükséges az, hogy illeszkedjenek egymáshoz az elemek. Ezeknek az alkalmazásoknak jól illeszkedniük, mert előfordulhat, hogy ezeket különböző részekre kell bontanunk a lehető leggyorsabban. (Desai, Fountaine, & Rowshankish, 2022)

Minél több visszajelzés érkezik és minél több ember képes ugyanazon információhoz hozzájárulni, annál jobb megoldások képesek születni az olyan problémákra, mint például a várakozási idő szállítás esetén, a biztonság, a karbantartás vagy pedig a munkahelyi beosztás. Fontos az, hogy egy vállalaton belül minél többen férjenek hozzá az adatokhoz és olyan válaszokat tudjanak adni a problémákra, hogy azok képesek legyenek bevételt termelni. (Desai, Fountaine, & Rowshankish, 2022)

Az adattermékek fejlesztését és kezelését olyan menedzserek felügyelik, akik a piaci igényeket kutatják és annak kielégítésének helyes marketingelésével foglalkoznak. Minden termék rendelkezik egy ilyen felelőssel és csapattal, akik felelősek annak legjobbra fejlesztéséért. Ezeket a csoportokat az adatelemzők, a modellezők, a mérnökök alkotják és számukra elengedhetetlen az, hogy hozzáférhessenek a működési, a folyamat béli, a jogi és a kockázati elemzésekhez. Közvetlen visszajelzéseket kapnak a végfelhasználótól a csapatok tagjai, ezáltal tudják pontosítani a termékekkel kapcsolatos igényeket. Az ügyfélközpontú rendszerek képesek vonzani a felhasználókat, mivel azt érezhetik, hogy törődnek velük és a véleményükkel ők is képesek fejleszteni a termékeket. A korai hozzáférésű produktumok is költségcsökkentést jelentenek, hiszen a tesztelők elmondják a véleményüket és még a megjelenés előtt tudják javítani a hibákat. (Desai, Fountaine, & Rowshankish, 2022)

Egy kiválósági központ segíti a termékek csapatait abban, hogy meghatározza a szabványokat és az eljárásokat. Emellett ők szabják meg a csapatoknak, hogy a dokumentációs folyamatnak, milyen módon kell zajlania, hiszen egy auditálás során ezt is ellenőrzik az adatminőség és piacra szánt termékekkel párhuzamban. Ezzel képesek a vállalatok a pazarlást és a bonyolultságot elkerülni. Ámbár sok vállalat már rendelkezik hasonló kaliberű részleggel, mégsem mindegyikük fektet rá kellő hangsúlyt. (Desai, Fountaine, & Rowshankish, 2022)

A termékeknek a teljesítményét mérni lehet és kell is. Erre különböző eszközök vannak mint, például eladások, elégedettség, nyereség. A felsoroltakon kívül, terméktől függően létezhetnek egyéb eltérő mérési számok is, ilyen például az aktív felhasználók száma.

Erre egy telekommunikációs vállalat esettanulmánya a legalkalmasabb. 150 felhasználó szokásait követték nyomon. Három év elteltével és az információk tudatában sikerült a cégnek több százmillió dollárt spórolniuk. (Desai, Fountaine, & Rowshankish, 2022)

A szalagról leérkező termékeket minőségellenőrzésen keresztül biztosítják, hogy azok a célnak megfelelően legyenek képesek működni. Ehhez az kell, hogy az adatdefiníciókat szigorúan vegyék és azok rendelkezésre is álljanak. A menedzserek érdeke, hogy az alkalmazottakkal együttműködjenek. Minőségbéli kár akkor keletkezik, ha egy adatot több rendszerben is tárolunk, ekkor duplikáció készül róla. Ezért szükséges az egyedi azonosítók használata. (Desai, Fountaine, & Rowshankish, 2022)

4. AZ ADATOK ÉS ADATBÁZISOK BIZTONSÁGA

4.1. Mi az adatbiztonság és miért fontos számunkra?

Az Oracle szerint ezt jelenti az adatbiztonság: „Az adatbiztonság olyan védelmi intézkedésre utal, amely az adatokat illetéktelen hozzáféréstől és bizalmas tulajdonságának megőrzése érdekében alkalmazandók. Az adatbiztonság legjobb gyakorlatai közé tartoznak az olyan adatvédelmi technikák, mint az adattitkosítás, kulcskezelés, adat szerkesztés, adat elfedés, kulcskezelés és a felhasználók hozzáférésnek ellenőrzése mellett az auditálás és felügyelet.” (What is Data Security?, 2020)

Az adat minden szervezetnek az egyik legfontosabb értéke. Mivel ekkora erővel bírnak az adatok, így szükséges őket megvédeni a nem engedélyezett hozzáférésektől. Az adatszivárgás és a sikertelen auditálások azt jelentik, hogy a szabályozásokat nem tartották be vagy pedig eltértek azoktól. Amennyiben egy ilyen esemény bekövetkezik, képes akár egy márka hírnevét is rombolni, mely után értékcsökkenés történik. Az ilyen esetek megelőzése érdekében született az Európai Unió általános adatvédelmi törvénye (rövidebben GDPR). Az adatszivárgásokkal kapcsolatban minden vállalatnak kötelessége azonnali hatállyal a hatóságok felé fordulni, és ezek be nem jelentése olyan következményekkel járhatnak, mint a vállalat éves bevételének 4%-os vagy pedig 20 millió euróig terjedő megbírságolása, attól függően, hogy melyik a magasabb összeg. Ha egy szűkebb körről van, akkor az érintett személyeknek jelentenie az incidenst. Azok minősülnek érzékeny adatnak, amikkel képesek vagyunk személyeket beazonosítani, továbbá pénzügyekkel, egészségüggyel vagy pedig a szellemi tulajdonnal kapcsolatosak. (What is Data Security?, 2020)

4.2. Hogyan védhetjük adatainkat?

Az adatmaszkolás, bontás és törés olyan technikák, amikkel tudjuk csökkenteni a különböző alkalmazásokban vagy adatbázisoknak megtalálható érzékeny adatok kitettségét, melyek fontosak a szabályozások szempontjából. A GDPR nagy erővel bíró szabályozás, ami olyan elvekre épül, mint a célhoz kötöttség, az átláthatóság és az integritás. A globalizáció és a digitalizáció miatt az Európai Uniónak új adatvédelmi elveket kell meghatároznia, mint például az elszámoltathatóság és az adatok minimalizációja. (What is Data Security?, 2020)

Az adatbiztonsághoz szükséges a megfelelő szintű technikai és szervezeti ellenőrzés, ami fontos az adatvesztés és az adatszivárgás szempontjából. A szabályozások nagyban ösztönzik a vállalatokat arra, hogy beleépítsék programjaikba a titkosításokat és a hálózati rendszerintegritást.

A felhasználók gyakorolják a legnagyobb felelősséget dokumentumaik felett, hiszen ők az egyetlenek, akik kiterjesztett adatvédelmi jogosultságokkal rendelkeznek. Amennyiben ezeket az adatokat más személy számára továbbítják, onnantól áll fenn az adatszivárgás lehetősége. (What is Data Security?, 2020)

A cégeknek évente többször is át kell esniük auditáláson. Az ellenőrzés leginkább az általuk tárolt és készített dokumentumokat és nyilvántartott biztonsági gyakorlatokat érinti. Az auditálások során, megvizsgálják a vállalatok adatbázisait, rendszereit és üzleti folyamatait, majd elemzik azokat, hogy biztosítani tudják az adatok integritását, azok biztonságát, illetve azt, hogy megfelelnek-e jogszabályoknak és irányelveknek. Az auditálás egy tervezési fázissal kezdődik, ami meghatározza az auditálás célját, a területet, amit vizsgálni fognak a folyamataival együtt és az időtartamát. Az ellenőrizni kívánt részből adatokat gyűjtenek különböző adatbázisokból és rendszerekből, majd azokat elemzik meghatározott szempontok alapján. Ezek az adatok lehetnek log, end-to-end folyamatokat leíró és munkavállalók tevékenységét tartalmazó dokumentumok. A fájlok alapján egy riportot készít az auditáló cég, ami tartalmazza az auditálás célját, módszereit, a kockázatokat, illetve a javaslatokat. Amennyiben észrevételeket tesz az auditor szervezet, a hibákat kötelességük korigálalni. Ajánlások esetén az érdekük az, hogy javítsák az adatbiztonságot, csökkentsék a kockázatokat és javítsák a problémákat. (What is Data Security?, 2020)

Az adatbázisok tartalmi vonzónak válnak az adathalászok számára, elsősorban az érzékeny és értékes információk miatt. Két fajtája van a tolvajoknak: külsős és belsős. A hackereket, kiberbűnözőket, bűnszervezeteket és nemzetek által támogatott szervezeteket sorolunk a külsős csoportba, nekik a céljuk, hogy zavart okozzanak a szervezet életében és globálisan egyaránt. A belsősök közé tartoznak azok a dolgozók, akik jelenleg vagy pedig előzőleg dolgoztak a vállalatnál, illetve azok, akik ügyfelek vagy pedig partnerek voltak. Akár külsős, akár belsős az illető, mindenképpen veszélyt jelent az adatokra és a szabályokra nézve. Módszereik egyediek, hogy képesek legyenek információkhoz jutni. A leggyakrabban használt támadási forma az e-mailen keresztül küldött vírusos link vagy szoftver, ami megfertőzi a rendszergazda gépét vagy egy olyan személyt, aki több jogosultsággal is rendelkezik. Sokszor hackerek vagy fejlesztők kihasználják az alkalmazások adta gyengeségeket. Ilyen például az SQL injekció (egy bizonyos kód beillesztése egy query felületre, ami egy folyamatot fog végrehajtani) vagy pedig a rendszer biztonsági részének megkerülése (ezt nem frissített vagy rosszul konfigurált adatbázisokon szokták használni). Fontos, hogy a biztonsági mentéseket vagy backup fájlokat is helyesen

tároljuk, mert ezek az archív dokumentumok is tartalmaznak információkat. A fejlesztés alatt álló projektek nem szoktak magas szintű védelem alatt állni, illetve nincsenek olyan szinten biztosítva, hogy minden támadást vissza tudjanak verni, mivel ezeket csak egy zárt körben alkalmazzák, nem úgy, mint azokat a fájlokat, amiket aktívan használnak. Vannak esetek, amikor nyílt irodában dolgoznak a munkavállalók és ekkor tudódhatnak ki érzékeny információk a akaratuk ellenére is, hiszen ilyenkor mindenki láthatja a monitorokat. A jelszómegosztás, a rossz konfiguráció, vagy esetleg egy baleset szokta a legnagyobb rést ütni a rendszer pajzsán, amelyet nagyrészt az emberi gondatlanság okoz. (What is Data Security?, 2020)

4.3. Az adatbiztonság gyakorlatai

Az eszközök rendszeres ellenőrzése hozzájárul az adatbázis biztonsági jóllétéhez, illetve fontos az, hogy képesek legyünk felismerni a konfigurációban történő változásokat. A szervezetek rendelkeznek egy alapbeállítással, hogy a közte és a mostani állapot között tudják vizsgálni az eltéréseket. Ezekkel az ellenőrzésekkel tudják elősegíteni az érzékeny adatok meghatározását a vállalatok. (What is Data Security?, 2020)

Folyamatosak az ellenőrzések arról, hogy a felhasználóknak milyen hozzáféréseik vannak alkalmazásokhoz és adatokhoz, és képesek megállapítani a gyanús viselkedéseket és ezzel kizárni a veszélyforrásokat. Ezek a megelőző vizsgálatok letiltják a hozzáférést titkosítással és maszkolással. A végső cél az, hogy ne kerülhessenek ellenőrizetlen kezekbe adatok. (What is Data Security?, 2020)

Az ellenőrzéseknek két módja van, az adatspecifikus és a felhasználóspecifikus. Az adatok által vezérelt ellenőrzés során alkalmazások szintjén figyelik a hozzáféréseket és a szabályokat, ezzel folyamatosan tudják több applikációban is figyelni és jóváhagyni a különböző kéréseket. A felhasználók szerinti vizsgálatoknál egy egyszerű elvet követnek, hogy az adott felhasználónak csak és kizárólag ahhoz az adathoz van hozzáférése, ami ténylegesen jóvá lett neki hagyva. (What is Data Security?, 2020)

Az adatbiztonsághoz szükséges tudatosabban és átgondoltabban megfigyelni a nagyobb vállalatok módszereit. Csökkenteni kell az adatszivárgás lehetőségét és egyszerűsíteni kell az adatbiztonsági gyakorlatokat, ebbe beleértve a titkosítást és a kulcsok kezelését, a maszkolást, a felhasználói hozzáféréseket és az aktív ellenőrzéseket is, melyek adatvédelmi szempontból is egyaránt fontosak. A felhasználók hitelesítése egy rendkívül fontos lépés, hiszen nem feltétlenül

férhet hozzá bárki bizonyos adatokhoz, emellett azt is tudnunk kell, hogy az adott személynek milyen jogosultságai vannak az adathalmaz szerkesztésére. Minden adatbázist aktívan kell figyelni, hogy milyen folyamatok mennek végbe a hálózaton és mik azok a lépések, amik elindítanak egyes automatizációkat. Ezeket csak úgy lehet visszakövetni, ha az adatbázis rögzíti azt, hogy mikor történt bejelentkezés és milyen kliensből küldte a felhasználó az engedélykérést, továbbá azt, hogy milyen SQL kód futott le. (What is Data Security?, 2020)

A felhőben tárolt adatbázisok előnyei közé tartoznak, hogy költséget tudunk velük csökkenteni, a munkavállalókról terhet vesznek le és fontosabb munkavégzési területekre tudják őket allokálni. Természetesen megvan a hátulütője annak, hogy ha házon kívülre helyezzük az adatbázisunkat, ekkor például megnövekszik a válaszidő a kliens és szerver között, vagy egy ismeretlen adminisztrációs jogosultsággal rendelkező csapatnak betekintési lehetősége lesz a cégbe. Ellenben, ha az adatbiztonság gyakorlásához a megfelelő munkavállalókat alkalmazzák, akkor a vállalatok képesek a költségeiket csökkenteni és növelni az agilitásukat. (What is Data Security?, 2020)

5. A FEJLESZTÉS

5.1. Az elképzelés

Az Attilával készített interjú sokat segített abban, hogy miként induljak el egy saját adatbázis elkészítésben. Első körben vázlatokat kezdtem tervezni az elhangzottak és a saját elképzelésem alapján. Mivel szeretek problémákat a legegyszerűbb módon megoldani, így a lehető legkevesebb, ám a feladat szempontjából elengedhetetlen táblával igyekeztem ezt is kivitelezni. Egy optimális tábla 10-20 oszloppal dolgozik, így ezeket a számokat nem haladtam meg, amennyiben szükséges volt, ott melléktáblákat iktattam be.

Számos nélkülözhetetlen, a vállalat alapvető működéséhez létfontosságú eleme dolgozói, a termék, az eladási, a rendelési és a beszállítói táblák. Az megemlítették alapján lehetőség van nyomon követni, hogy melyik eladó milyen terméket értékesített, valamint meghatározni, hogy mely árukkal realizálódik nagyobb forgalom az adott időszakban, így az üzletvezetők könnyebben azonosíthatják a gyorsan és nagyobb mennyiségben fogyó cikkeket.

A táblák tulajdonságait érdemes nem ismételni, mivel a duplikációk problémát okozhatnak az adatbázisban, e miatt célszerű a műveleteket inkább melléktáblába rendezni.

5.2. A vállalat SGM táblája

A vállalat SGM táblája nem kötelező eleme annak, hogy egy adatbázist el tudjunk készíteni, ennek ellenére kiváló ellenőrző eszköz, amely segít elkerülni a hibákat a tényleges folyamat során. Az Iguana.pub SGM tábláját a vázlataim alapján tudom a legegyszerűbben elkészíteni. Ugyan a tulajdonságok nevei gyakran ismétlődnek, nem minden táblába kerülnek be. Az adatbázis az SGM-en alapul, de időközben később akár ki is egészülhet több táblával, amiket nem tüntetek fel. „Az elképzelés” bekezdésben leírt sorrendet fogom követni, így például a dolgozó táblához a dolgozói azonosító, a vezeték- és keresztnév, a születési hely- és dátum, az irányítószám, a lakcím, a tájszám, az adószám, a bankszámlaszám, a telefonszám és az email kerül. A többi táblát is hasonló elgondolással építem fel. Az adatbázis működése miatt fontos, hogy ezen elemek képesek legyenek egymással csatlakozni, amit úgy érhetünk el, hogy létrehozunk közös pontokat. Amennyiben egy sorban kettő vagy attól több jelzés látható (legyen az csillag, kör vagy nyíl), akkor kettő vagy több tábla kapcsolatban áll egymással (2. ábra).

rendeltem hozzá, mivel ezekkel nem vagyunk képesek meghatározni, hogy melyik entitásról is beszélünk. Például Kovács Attilából lehet egy vállalatnál három, de Kovács Attilából, akinek 5-ös azonosító kódja van, csak egy lehet fel a cégnél vagy a szóban forgó adatbázisban. A szabály továbbra is érvényben van: egy oszlopban nem szerepelhet egyszerre egyszerű és- részazonosító. A részazonosítók azokba a cellákba kerültek, ahol egy táblán belül több azonosító is kapcsolódik egymáshoz.

5.3. A program és annak nyelvezete

A feladat megvalósításához használt szoftver az SQL Server Management Studio (SSMS), ami még egyelőre csak Windows operációs rendszeren képes futni, de Linuxon és macOS-en a platformok összekötéséhez megfelelhet az Azure Data Studio is. A program egy integrált környezet, amely azt a célt szolgálja, hogy adatbázisokat tudjunk létrehozni és kezelni őket, továbbá bármelyik szempontból vizsgálva kielégíti a felhasználói igényeket. Az SSMS grafikai eszközöket és script szerkesztőket kínál felhasználóknak, ezzel segítve őket, hogy minden szinten képesek legyenek kezelni a felületet és az adatbázisokat. Mivel tanulmányaim során a MySQL-t használtam, így a megtanultakat az SSMS szoftverben használt Transact SQL programozási nyelvre kell most átfordítanom. Sokban a két programozási nyelv ugyan nem tér el egymástól, viszont meg kell szokni az új környezetet. (What is SQL Server Management Studio (SSMS)?, 2023)

Az SQL egy standardizált nyelv, ami leginkább a relációs adatbázisok kezelésre szolgál. Az adatbázis kezelőkben képesek vagyunk lekérdezéseket és módosításokat elvégezni parancsokkal. Sokkal inkább arra használjuk ezt a nyelvet, hogy egy adott célt elérjünk anélkül, hogy megmondanánk hogyan történjen az. A Transact SQL pedig a Microsoft által készített bővítmény a nyelvezethez, mellyel a Microsoft SQL Server-ben és a Sybase ASE-ben találkozhatunk. Maga a nyelv ugyanazt a struktúrát tartalmazza, szimplán addicionálisan kapunk programozási konstrukciókat, változókat, támogató függvényeket, amikkel képesek vagyunk manipulálni a dátumokat és az időt. Emellett találkozhatunk a hibaelhárítás és kezelés különböző formáival, áramlásvezérlés nyelvvel és tranzakció támogatással is. (Transact-SQL reference (Database Engine), 2023)

A táblákban látható tulajdonságok nevei mellett adat típusok helyezkednek el. Ezeknek különböző jelentőségük van attól függően, hogy melyiket választjuk ki. Az adatok típusai meghatározzák azt, hogy mekkora helyet fognak elfoglalni az adatbázisban és mekkora a terjedelmük, melyek változnak csoportonként is. Az adatbázis tárhelyigénye függ attól, hogy mennyi adatot tárolunk

benne és attól is, hogy milyen az adat típusa. Például az egész számokat az SQL és még sok más programozási nyelv `int`-nek, azaz `integer`-nek hívja. A `transact SQL`-ben létezik `tinyint`, `smallint`, `int` és `bigint`. Ezeknek az egész számoknak különböző tartományaik vannak például a `tinyint` 0 – 255-ig és a `bigint` -9.223.372.036.854.775.808 – 9.223.372.036.854.775.807-ig terjed. Látható, hogy a két intervallum között hatalmas a különbség és ez megmutatkozik tárhely szempontjából is, tehát míg az előbbi 1 byte-ot, addig az utóbbi 8 byte helyet foglal el. Az elvégzendő művelet sebessége az adathordozón foglalt hely és az adatbázis nagyságától függően változik, tehát kisebb adatbázisban sokkal gyorsabban mennek végbe folyamatok, mintha ezt egy több adattal rendelkezővel szeretnénk lefuttatni; pontosan e miatt fordulhat elő a teljesítménybeli különbség. (Data types (Transact-SQL), 2023) (`int`, `bigint`, `smallint`, and `tinyint` (Transact-SQL), 2023)

Érdekes már az adatbázisunk kiépítésekor arra törekedni, hogy olyan adat típusokat válasszunk a feladathoz, ami az adott célnak megfelel. Az előző egész számos példára visszatekintve, ha egy dolgozóknak szeretnénk azonosítót adni és `tinyint`-ként meghatározni, akkor csak 254 dolgozót fogunk tudni felvinni a rendszerbe; ez a fluktuációtól, a cég méretétől és létezése óta eltelt időtől függ. Amennyiben nagy a fluktuáció a vállalatnál, számíthatunk arra, hogy 3-4 éven belül újra kell osztani az azonosítókat a rendszerben, esetlegesen át kell dolgozni az egészet. Így a legcélszerűbb, ha azonnal a `smallint` vagy pedig az `int`-et választjuk opciónak. Több helyen is tudunk hasonló módon dönteni, ilyen előfordulhat például a személyi okmányok estén, melyek általában fix karakterszámmal rendelkeznek. Ebben az esetben célszerű `nchar`-t használni, így már fixált byte mennyiséget fog használni a cella. Ha tájszámról beszélünk, akkor a karakterszám egy 9 byte-ot foglaló fix rekord lesz. Az `nvarchar`, mint neve is részlegesen említi, egy variálható méretű adat. Ezt a fajta a típust sokszor előnyben részesítik, hiszen igazodni tud azokhoz az adatokhoz, amik eltérő hosszúságúak, mellyel sok esetben tárhelyet lehet megspórolni. Ezekre a részletekre érdemes odafigyelni, mert egy adott karakterszámnál nem tudunk bővíteni, így ekkor válik relevánssá a variálható karakterhossz rugalmassága, ezért inkább több variálható megadása szükséges, nehogy törölni kelljen a táblát emiatt. Az `nchar` és `nvarchar`-hoz muszáj értéket rögzítenünk, ezt egy 1 – 8000 közötti skálán tudjuk megtenni (ha az `nvarchar`-t használjuk, akkor ott lehet a „max” kifejezéssel is élni, melynek a maximális tárolókapacitása 2 gigabyte). (`char` and `varchar` (Transact-SQL), 2023)

5.4. Táblák, kapcsolatok és kulcsok hozzárendelése az adatbázishoz

Az SSMS felhasználói felülete egyszerű és könnyen értelmezhető. Dolgozatom témájának bemutatása érdekében létrehozott adatbázist „Iguana.pub_database”-nek neveztem el. A felépítési folyamatot a dolgozók táblával kezdtem el, melyhez egy Query felületet nyitottam meg, ahová a kódot kellett begépelnem. Először is ki kellett választanom, hogy melyik adatbázist szeretném használni, erre a „USE” parancsot kellett alkalmazni, majd ezt követte az adatbázis neve és egy „GO”. Amint kiválasztottam az adatbázist, beillesztettem a következő kódot: DROP TABLE IF EXISTS Dolgozo. Ez hasznosnak bizonyult számomra, hiszen a későbbiekben előfordult olyan eset, hogy be kellett szűrnöm egy új tulajdonságot a táblába, azonban ehhez előtte törölnöm kellett a már korábban létezőt. Ennek egyszerű magyarázata az, ha már létezik egy adott eleme az adatbázisnak (lásd Dolgozó tábla), akkor nem tudom létrehozni és felülírni ugyanazt, még akkor sem, ha nem egyeznek meg tartalmilag.

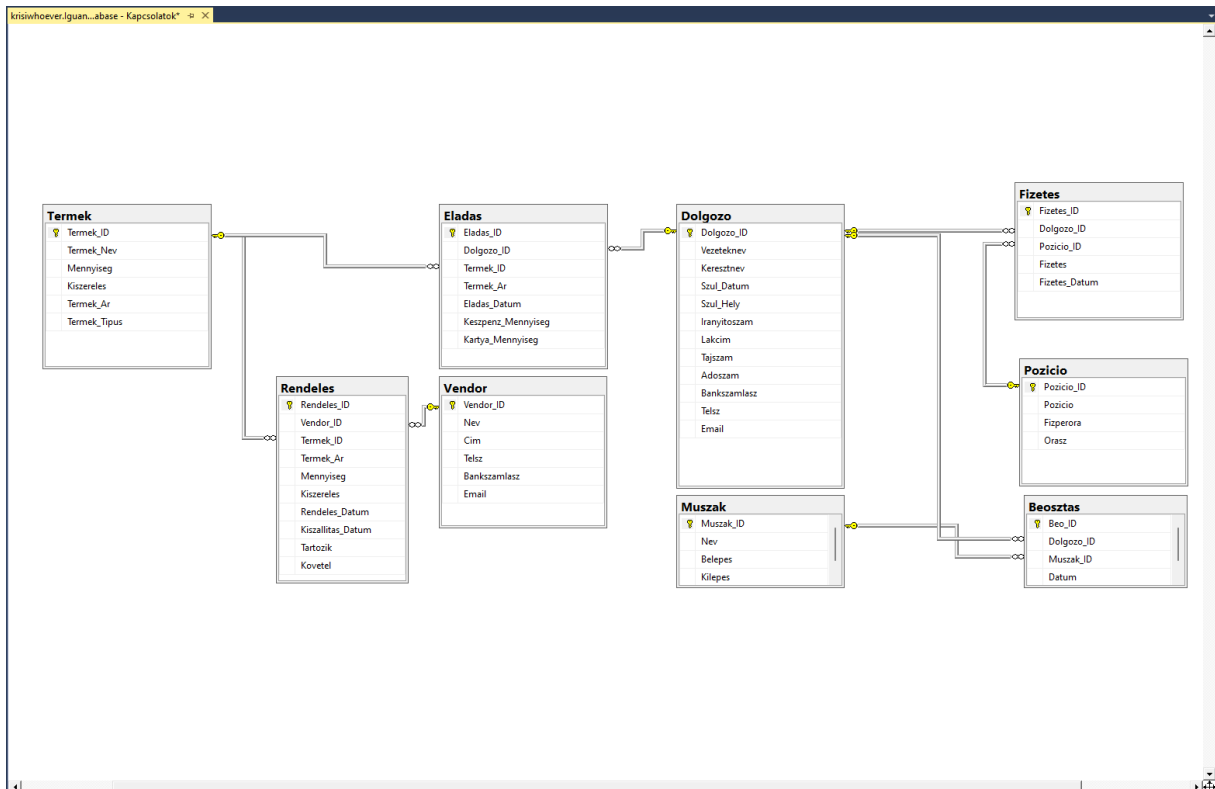
A következő lépés a „CREATE TABLE” volt, ez hozta létre magát a táblát, de kellettnek hozzá tulajdonságok is. Egy tulajdonság három fontos részből tevődik össze: a tulajdonság rövid neve, az adat típusa és az, hogy egyáltalán megengedett-e a cella üresen hagyása. Példaként levezetem a dolgozó tábla létrehozását. A táblában a már korábban látott tulajdonságok jelennek meg és ehhez kellett hozzáigazítanom az adat típusát, elsősorban az azonosítót. Hogy ne kelljen a közeljövőben újra osztani a dolgozókat az int-et használtam, és ez bőven elég is egy kis vállalat számára legalább egy évtizedre. A vezetéknevet és keresztnévet nvarcharral használtam, hiszen minden embernek különböző hosszúságú a neve és ezzel sokkal hatékonyabban lehet kihasználni a tárhelyt, továbbá nem fog megnövekedni a lekérdezések ideje. A születési dátumhoz date argumentumot választottam, ezzel pontos dátum formátumot tudtam hozzárendelni. Születési helyhez, e-mailhez és lakcímhez szintén nvarchar került, viszont a következőkben szigorúan csak nchar került be a tulajdonságok mellé, hiszen az irányítószám, tajsám, adószám, telefonszám és bankszámlaszám fix értékkel rendelkeznek. Mindegyik tulajdonsághoz not null kikötést alkalmaztam, kivéve a bankszámlaszám esetén, mert nem mindegyik személy rendelkezik bankszámlával, így ott megengedtem azt, hogy esetlegesen üres cella legyen az adatbázisban. A többi táblával hasonló módon jártam el, megvizsgáltam a legoptimálisabb típusok lehetőségét.

Az összes tábla elkészültekor hozzárendeltem az azonosító kulcsokat. A dolgozó táblában a dolgozó azonosítójához rendeltem a kulcsot, a többi táblában is hasonló módon jártam el, szigorúan csak az azonosítók kaptak kulcsot.

A kapcsolatokat a táblák között (3. ábra) úgy határoztam meg, hogy ahol egy tulajdonság csak leíró volt, az egy másikban táblában idegen kulcsként jelent meg (amennyiben mindkét táblában jelen volt az adott tulajdonság, lásd: azonosítók). Az SGM táblában remekül látszik, hogy melyik tulajdonság milyen szerepet vett fel éppen.

3. ábra

A táblák közti kapcsolat



Forrás: saját szerkesztés

5.5. Táblák feltöltése adatokkal a Mesterséges Intelligencia segítségével

Sikeresen létrehoztuk az adatbázist benne táblákkal, kulcsokkal és kapcsolatokkal. Ahhoz, hogy képesek legyünk riportokat vagy lekérdezéseket készíteni, szükséges volt a táblákat a következő módon feltölteni. A már látott „USE” parancs segítségével egy Query Editor-ban kiválasztottuk az általunk létrehozott adatbázist: „USE [Iguana.pub database] GO”. Miután sikeresen használatba vettük az adatbázisunkat, az „INSERT INTO” segítségével beillesztjük a számunkra releváns adatokat meghatározott táblákba és cellákba. Fontos, hogy az adatok, amik beillesztésre kerültek, nem haladhatják meg megadott karakterszámot (lásd: nchar és nvarchar esetében). Az „INSERT INTO” -t a tábla neve, majd a tulajdonságok rövid neve követte abban a sorrendben, ahogy azt a táblában már beállítottuk. Ezek után egy „VALUES” kifejezés került a következő sorba, ezzel jelezve

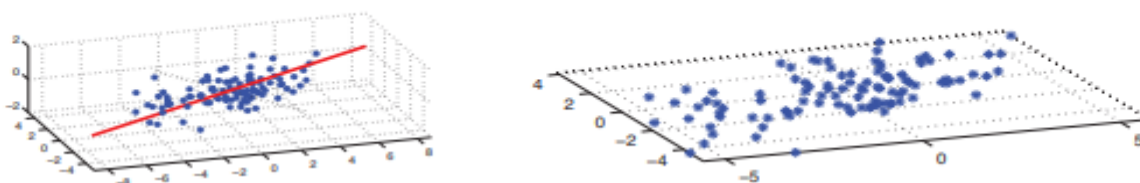
azt, hogy az alábbi adatok kerülnek felvételre. Az adatoknak követnie kellett ezt a formulát, hiszen, ha az adat nem megfelelő formátumú és nem a helyes sorrendben helyezkedik el, akkor a rendszer programozási nyelvhibát jelezett volna számunkra.

Bizonyos táblákat időigényes lett volna kézzel feltölteni (ilyen például a Termék tábla), így a mesterséges intelligencia segítségével rövidítettem meg ezt a folyamatot. De mi is az a mesterséges intelligencia és miért is releváns számunkra ebben a témakörben? A mesterséges intelligenciának több fajtája van: a gépi tanulás (machine learning), a mély tanulás (deep learning), a természetes nyelvi feldolgozás (natural language processing), a gépi vizualizáció (computer vision), az etikus és a társadalmi vonatkozások (ethical and societal implications). Ebből számunkra a két legfontosabb típus a gépi és mély tanulás, ugyanis ezekkel találkozunk a leggyakrabban és ezek illettek bele az általam kutatott témakörbe.

A gépi tanulás egy olyan algoritmus, amely képes tanulni adatokból. „Egy számítógépes programnak azt mondják, hogy tanuljon a tapasztalatokból (E) a feladatok bizonyos csoportja tekintetében (T) és a teljesítmény méréséből (P). Ha a gépi tanulás teljesítménye mérve van (P) a feladatokban (T), akkor fejlődik tapasztalatokból (E)”. (Goodfellow, Bengio, & Courville, 2015) Naggy általánosságban a gépi tanulási rendszer példákban tanul, ennek szemléltetésére vegyünk egy hangulat vizsgálatot. Amennyiben több fajta emócióról készítünk képeket (minél többet, annál pontosabban fogja tudni a gép megállapítani az adott hangulatot), akkor képes lesz a gép behatárolni azt, hogy egy bizonyos személy éppen milyen hangulatban van. Ha egy adott emóció nincsen „regisztrálva” (nincsen róla ismerete), akkor valamelyik általa már létező csoportba fogja kategorizálni, amihez szerinte a legközelebb áll (ezt mind vektoroknak az elhelyezkedéséből állapítja meg). Természetesen az eredményt lehet pontosítani különböző módszerekkel, például, ha a tanulás felügyelve van, de vannak esetek, amikor a gépet saját magára is lehet hagyni és ellenőrizetlenül folytatja a tanulást. Abban az esetben, amikor predikatív módszert alkalmazunk, a cél az, hogy az általunk bevitt (input) információkból, adatokból tanuljon és képes legyen azokból egy összekapcsolást létrehozni, majd eredményt (output) biztosítani számunkra. Amennyiben képes a rendszer kategorizálni egy problémát vagy feladatot, akkor klasszifikációról vagy más néven, mint a felismerésről beszélünk, ha pedig igazi eredményt kapunk, akkor regresszióról van szó. A felügyelet nélküli tanulást alatt azt értjük, hogy egy eredményt kapunk anélkül, hogy bevinnénk bármilyen adatot a rendszerbe. A leggyakoribb feladat ebben az esetben csoportokba való klaszterezés.

Gyors összehasonlítás: veszünk 210 főt és őket egy két-dimenziós ábrán fogjuk reprezentálni (3. ábra). Előfordulhat, hogy többfajta klaszterezés vagy alcsoport jelenik meg, de nem tudjuk pontosan, hogy mennyi. Elsők között meg kell állapítanunk, hogy a mennyiségnek milyen eloszlása van a klaszterek között. Amikor a felügyelt módszert alkalmaztuk (4. ábra bal oldala), feltételezzük, hogy két csoport volt – férfiak és nők – de, ahol nem volt felügyelet (4. ábra jobb oldala) ott szabadon választhattunk annyi csoportból, amennyiből csak szerettünk volna. A példa mellett más feladatokat is képes ellátni a mesterséges intelligencia, de az említettel lehet a legjobban szemléltetni azt, hogy egy adatbázisból és adatok sokaságából dolgozik. (Murphy, 2012)

Lineáris regresszió és Klaszterezés kinézete



Forrás: Kevin P. Murphy, 2012

A feladat, amit a mesterséges intelligenciára bíztam az volt, hogy adott sorrendben töltsse fel nekem az általam létrehozott adatbázis tábláit. Természetesen ezt külön-külön táblánként adtam ki parancsban, hogy ezzel csökkentsem a hiba lehetőségek számát. A gép által közölt eredményeket beillesztettem a már fentebb említett „INSERT INTO” függvénybe; a kód helyenként sikeresen lefutott, de voltak esetek, ahol először meg kellett bontanom a táblák között már létező kapcsolatokat, ugyanis hibát írt ki. Amikor az adatok beillesztésre kerültek sajnos már nem volt lehetőségem újra létrehozni a már megbontott kapcsolatokat a táblák között, mert az az ACID elveknek ellentmondott. Azokban a táblákban, ahol az „Identity” kód szerepelt, ott az ID hozzárendelése megfelelt, de más táblákban, ahol manuálisan kellett kitölteni az azonosító helyét, már problémába ütköztem, hiszen a gép nem 1-es számmal kezdte az azonosítást és az általa használt kódban sem említette az „Identity”-t, mint opció. Ebben a szituációban az Eladás és Dolgozó táblák közt eltérés volt az azonosítók tekintetében. A legelső eladási tranzakcióhoz 1-es azonosító és 1-es ID-val rendelkező dolgozó került. A mesterséges intelligencia által hozzáadott adatok között nem volt jelen 1-3-ig azonosítás a Dolgozó táblában, így a legelső névhez 4-es szám lett hozzárendelve. Tanulsággal zárult tehát a feladat megoldásának ezen része. Mindenképpen ellenőrizni kell az MI munkáját, mert nem feltétlenül azt az eredményt fogja számunkra biztosítani, mint amit mi elképzeltünk. Megvan az előnye és hátránya is annak, hogy minél inkább automatizálni és egyszerűsíteni szeretnénk az életünket, de a mostani formájában csak egy gépre bízni magunkat nem érdemes.

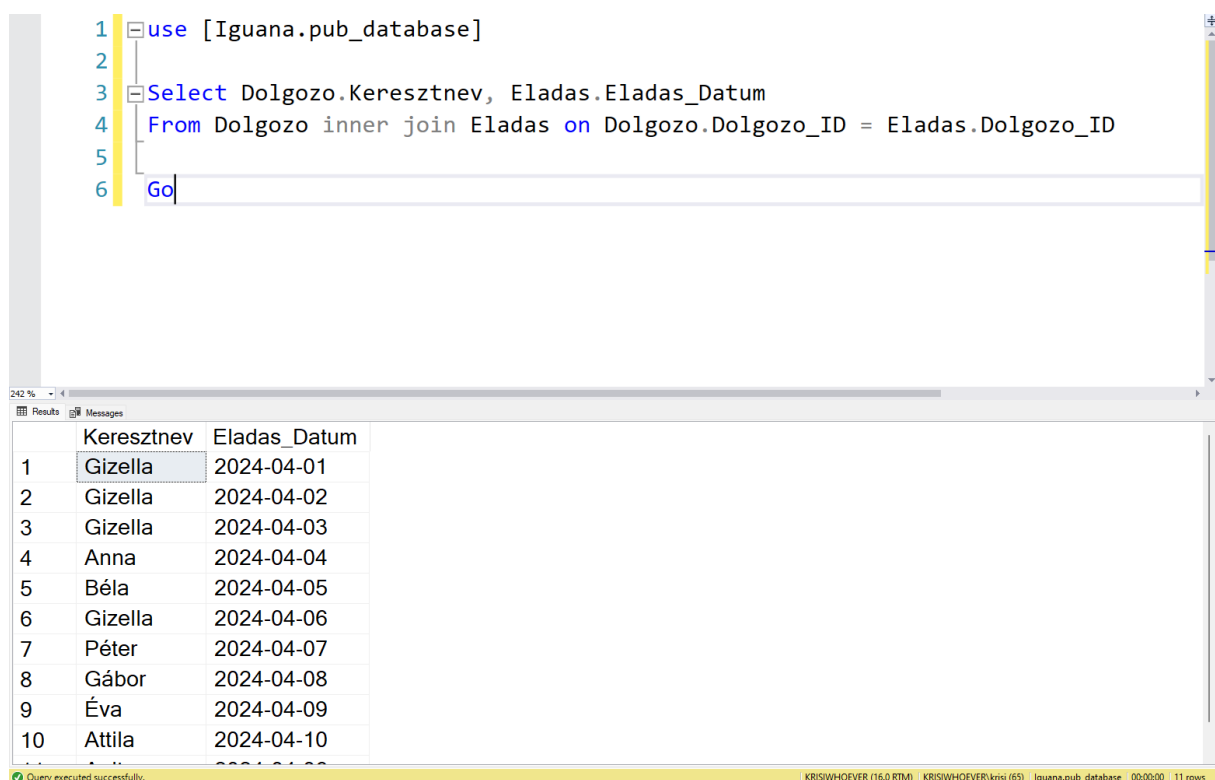
5.6. Lekérdezések és működőképesség

Az elkészült adatbázis innentől működőképes és alkalmazható különböző műveletekre. Hogy ezt bebizonyítsam demonstrálnám példákkal.

Tegyük fel, hogy listázni szeretnénk a dolgozók nevét, és azt, hogy mikor adtak el termékeket az eladási dátumok alapján. „USE” kóddal használatba vesszük az adatbázisunkat, majd egy „SELECT” paranccsal kijelöljük azokat a mezőket, amiket szeretnénk a lekérdezésünkben látni. Ebben az esetben a dolgozó táblából származó keresztnévet és az eladás táblából szükséges dátumot hívjuk meg, ezzel a sorunk a következő módon fog kinézni: „SELECT Dolgozo.Keresztnev, Eladas.Eladas_Datum”. Ha a kódot ezek után futtatnánk le, akkor sajnos hibát kapnánk, hiszen nem határoztuk meg azt, hogy mely táblák kapcsolódjanak egymással. Ezért a „FROM” paranccsal kell folytatnunk a megoldást egy „inner join” fogalommal társítva, ami magáért a csatlakozásért felelős. Tehát a következő sorban található kód: „FROM Dolgozo inner join Eladas on Dolgozo.Dolgozo_ID = Eladas.Dolgozo_ID;”. Folytatásképp, egyenlővé kell tennünk a két táblában található dolgozói azonosítókat, hiszen ugyanazzal az értékkel bírnak és képesek a már korábban említett táblák közötti kapcsolatot is a dolgozó ID által egymással kommunikálni. Az utolsó sorba a „GO” kerül, majd lefuttatáskor az alábbi eredményt kapjuk (5. ábra).

5. ábra

Eladók eladásai dátummal lekérdezés



The screenshot shows a SQL query editor with the following code:

```
1 use [Iguana.pub_database]
2
3 Select Dolgozo.Keresztnev, Eladas.Eladas_Datum
4 From Dolgozo inner join Eladas on Dolgozo.Dolgozo_ID = Eladas.Dolgozo_ID
5
6 Go
```

Below the query editor, the results are displayed in a table with the following columns: Keresztnev and Eladas_Datum. The results are as follows:

	Keresztnev	Eladas_Datum
1	Gizella	2024-04-01
2	Gizella	2024-04-02
3	Gizella	2024-04-03
4	Anna	2024-04-04
5	Béla	2024-04-05
6	Gizella	2024-04-06
7	Péter	2024-04-07
8	Gábor	2024-04-08
9	Éva	2024-04-09
10	Attila	2024-04-10

The status bar at the bottom indicates: Query executed successfully. KRISWHOEVER (16.0 RTM) KRISWHOEVER(krisi) (65) Iguana.pub_database 00:00:00 11 rows

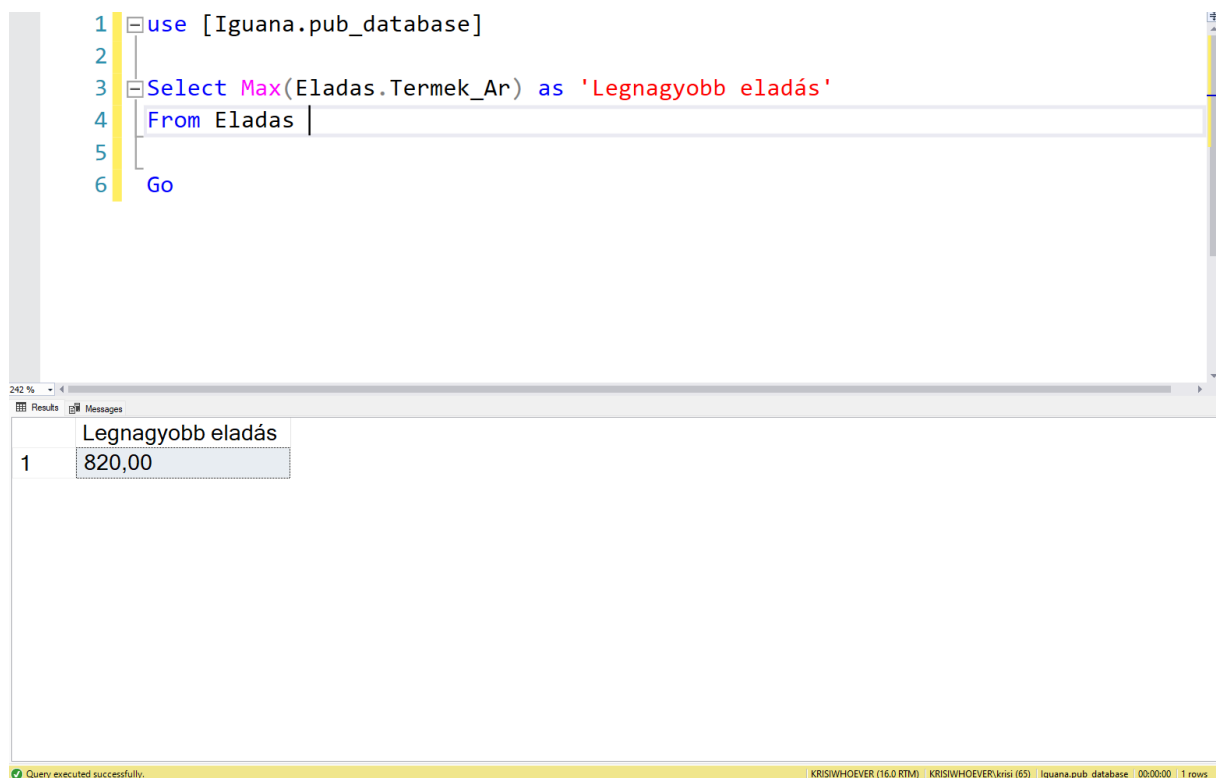
Forrás: saját szerkesztés

Ezzel sikerült listáznunk a dolgozók eladásainak dátumait. Lehetséges olyan egyéb műveleteket is végrehajtani, amivel ki tudjuk mutatni a legdrágább eladásunkat.

Az előző feladathoz hasonlóan használjuk az adatbázisunkat a „USE” paranccsal, majd ezt követi a „SELECT MAX”, amivel képesek vagyunk a legnagyobb értékkel rendelkező rekordot kikeresni. (6. ábra)

6. ábra

Legnagyobb eladott értékű áru lekérdezés



```
1 use [Iguana.pub_database]
2
3 Select Max(Eladas.Termek_Ar) as 'Legnagyobb eladás'
4 From Eladas
5
6 Go
```

	Legnagyobb eladás
1	820,00

Query executed successfully. KRISWHOEVER (16.0 RTM) KRISWHOEVER(krisi (65) | iguana.pub_database | 00:00:00 | 1 rows

Forrás: saját szerkesztés

A művelettel megkaptuk a legnagyobb eladási értékünket, ami 820 forint volt. Természetesen lehet ettől komplexebb műveletek is végrehajtani, de ezekkel a példákkal kiválóan lehet reprezentálni azt, hogy a kapcsolatok működőképeseek, továbbá, hogy léteznek rekordok is a táblákban. Persze ezekkel a parancsokkal még csak a felszínt kapargatom, de a megfelelő időbefektetéssel és tapasztalattal már bonyolultabb procedúrákat is képes véghez vinni az ember.

5.7. Késztermék

Az elkészült adatbázist a megfelelő helyre kell szállítania a fiktív vállalatnak. Ez olyan momentuma a szervezésnek, amiről a vezetőségnek kell döntenie, amennyiben egy nagyobb szervezetről

beszélünk, azonban egy kisebb cég életében szintén meghatározó az, hogy mégis hová kívánják telepíteni a készterméket. Az adatbázis szervert általában a vállalat irodájában szokták elhelyezni, vagy pedig az eszköz (vagy fájl) egy szerver parkba kerül. Az utóbbi egy kényelmi funkció a cég vezetői számára, hiszen így nem kell tartaniuk attól, hogy áramkimaradás során véletlenül leállna szerver és az adatok megsemmisülnének (abban az esetben, ha ez mégis bekövetkezne, a szerver még mindig elérhető otthonról). Az adatbázist egy ilyen kisbirtok szempontjából azért nem érdemes szerver parkba lokalizálni, mert sokkal gyorsabban képes elérni a szervert a kassza, ha fizikális kapcsolatban tudnak állni egymással. Az adatbázis legutolsó mentése egy merevlemezre került, majd azt a kívánt helyre való beillesztésének pillanatától kezdve, működőképessé vált maga a rendszer is.

6. ÖSSZEGZÉS

A 60-as évektől napjainkig az adatbázisok és a számítástechnika rengeteget fejlődött és ez a tendencia évről évre egyre jobban növekszik. Különböző fajta adatbázis modellek jelentek meg az idő előrehaladtával kezdve, a relációs adatbázistól egészen az adattárházakig. A közös bennük az, hogy ACID elven működnek, ettől lesz egy adatbázis felépítése megkérdőjelezhetetlen.

A 21. században körül vesz minket az adat és adatbázisok sokasága. Napról napra egyre több és nagyobb mennyiségű adatot termelünk, ezzel növelve az ökológiai lábnyomunkat, de a technológia fejlődésének segítségével képesek vagyunk különböző vívmányokkal (megújuló energia, energiatakarékos eszközök és kisebb méretű adathalmazok használata) ezt csökkenteni.

Napjainkban egy vállalat számára biztos alapot jelent egy olyan adatbázis, ami a cég működésének megfelelő módon van felépítve a saját igényei szerint. A cégeket olyan módszerekkel vagyunk képesek támogatni, amiket egy évtizede még el sem tudtunk képzelni. Már csak a számítógépek használata is nagyban hozzájárul ahhoz, hogy kulcsfontosságú döntéseket legyünk képesek hozni. A mesterséges intelligencia megjelenése az utóbbi években új kapukat nyitott meg, ami egyúttal új feladatokat tár az emberiség elé és ezt meg kell tanulnunk kezelni. Az MI által kapott lehetőségeket nem szabad elszalasztani, hanem azzal együtt kell tudnunk fejlődni, így több adattal, gyorsabban, sokkal pontosabb számításokat fogunk végezni. Nem szabad elfeledni, hogy nem a gépeknek kell döntenie az adott kockázatokról, hanem az ember lesz az, aki érzelmi és értelmi attitűdöt figyelembe véve fogja kimondani az utolsó szót és tud felelősséget vállalni.

A cégek leginkább olyan adatbázis kezelő rendszereket választanak, amik felhasználó-barátak, egyszerűek és letisztultak, hogy ezeket könnyedén tudják kezelni a vállalaton belül a munkatársak. Ezzel kapcsolatban szoktak és érdemes is továbbképzéseket tartani, hogy a munkavállalók sokkal tudatosabban és biztonságosabban tudjanak tájékozódni az alkalmazáson belül anélkül, hogy az irányelveket és szabályokat megszegnék. A kezelendő rendszert fontos úgy kiválasztani, hogy az a szervezet számára megfelelő legyen, így amennyiben egy multinacionális vállalatról beszélünk, akkor érdemes már egy felhő alapú adatbázis kezelőt alkalmazni, hogy azt mindenki el tudja érni. Annak ellenére, hogy válaszdőben ez egy lassabb folyamat, mint a törzshelyen tárolt adatbázis, több cég, mégis ezt az opciót választja a kényelem miatt, mert egy külső vállalat tartja karban a rendszert és felel a működéséért.

Amennyiben a szervezetek képesek alkalmazni a mostani trendek iránymutatásait, egyre jobb kimutatásokat tudnak készíteni. A megújuló energiák, az tördelése és archiválása, illetve az adatmennyiségek kicsinyítése segítik azt, hogy az ökológiai lábnyomunk csökkenjen. Az automatizáció és a mesterséges intelligencia bevonása egy cég életében sorsfordító lehet, de ezeket csak és kizárólag felügyelet mellett szabad alkalmazni és ellenőrizni az általuk elvégzett feladatokat. Ezek az eszközök inkább segítségként szolgálnak, mintsem munkaerő helyettesítésként, ezért szükséges az, hogy ez beépüljön a cégek gondolkodás módjába is. Továbbá olyan jótékony hatásai is vannak a vállalatokra nézve a fenti gyakorlatoknak, mint a költségeik optimalizálása, a kezelt adataik biztonságban tartása, valamint a riportok elkészítési idejének lerövidítése.

Az adatbiztonság egyaránt fontos a fogyasztó és a vállalatok számára, hiszen kölcsönös bizalmon épül a kapcsolatuk. Amennyiben egy szervezet nem képes biztosítani a vásárló vagy felhasználó számára az érzékeny adatainak a megfelelő körülményeket, súlyos következményekkel járhat, mint például elpártolás, jogszabályok megsértése (amivel együtt komoly pénzbeli büntetésekre számíthat a cég). Ezen problémák elkerülése érdekében folyamatosan ellenőrzéseket kell tartani, és ehhez különböző módszereket kell alkalmazni, mint titkosítás, maszkolás, hozzáférések állandó felügyelete. Erre célszerű a saját vállalatán belül alkalmazottakat keresni, akik ezzel foglalkoznak, vagy pedig, ha egy felhő alapú szolgáltatást veszünk igénybe, akkor a megfelelő szakembereket kell hozzá megtalálni, akik ezeket a feltételeket tudják biztosítani.

Az általam létrehozott adatbázisnak a megtervezése egy ebben már jártas személy interjúztatásával kezdődött, aki rengeteget tudott segíteni nekem a vállalat elképzelésében. Az SGM tábla létrehozásával előre tudtam tisztázni magamban bizonyos táblákat, milyen tulajdonságokat tartalmazzanak és milyen kapcsolatban álljanak egymással. Mikor digitalizálni kezdtem az adatbázist, tisztulni kezdett a kép, hogy több adat is felmerülhet a rendszerben és nem csak azok, amikre először gondoltam. Ehhez természetesen az is kellett, hogy elegendő időt töltsék magával a feladat megoldásával. A kapcsolatok létrehozása után a táblák adatokkal való feltöltése sajnos már az mesterséges intelligencia adta lehetőségekkel nem volt kompatibilis helyenként, hiszen az MI nem tudta feltétlenül, hogy milyen kódot alkalmaztam (például az IDENTITY 1,1), ezért a kód ACID elveknek mondott ellent. A lekérdezések segítségével teszteltem és prezentáltam az adatbázis helyesen való működését.

A hipotéziseim megválaszolásra kerültek, mégpedig azzal, hogy a vállalatok többsége azért preferálja a relációs adatbázisokat, mivel azokat képesek könnyen skálázni és egyszerűen kezelhetőek. Ugyan sok előnye van az adatbázis alapú rendszerek használatának, de ezeket csak akkor tudjuk optimálisan kihasználni, ha nem csak a létrehozásuknál vagyunk körültekintőek, de hangsúlyt fektetünk a későbbiekben a karbantartásukra is (így a költséghatékonyság mellett a teljesítménynövelés is fókuszba kerül). A felhő alapú adatbázisok nagy érdeklődésnek örvendenek, hiszen rengeteg felelősséget tudnak más szervezetekre áthárítani, mint például a szerver karbantartása, frissítése és ellenőrzése, melyekkel a saját munkaerejüket újra tudják allokálni munkahelyen belül.

IRODALOMJEGYZÉK

- char and varchar (Transact-SQL)*. (2023. 05 23). Letöltés dátuma: 2024. 04 09, forrás: learn.microsoft.com: <https://learn.microsoft.com/en-us/sql/t-sql/data-types/char-and-varchar-transact-sql?view=sql-server-ver16>
- Choubey, S. (2023. 05 09). *Gartner*. Forrás: gartner.com: <https://www.gartner.com/en/newsroom/press-releases/2023-05-09-gartner-identifies-the-top-ten-data-and-analytics-trends-for-20230>
- Data types (Transact-SQL)*. (2023. 05 23). Letöltés dátuma: 2024. 04 09, forrás: learn.microsoft.com: <https://learn.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16>
- Desai, V., Fountaine, T., & Rowshankish, K. (2022. 08). *Harvard Business Review*. Forrás: hbr.org: <https://hbr.org/2022/07/a-better-way-to-put-your-data-to-work>
- Dr. Budai, L. (2020). *Az adattól az adatbázisig: adatmodellek, adatstruktúrák, strukturált és nem strukturált adatok kezelése, relációs adatbázisok*. Dr. Budai László, Budapest, Pest, Magyarország.
- Fogalomtár*. (2014. 07 30). Letöltés dátuma: 2024. 03 17, forrás: web.archive.org: https://web.archive.org/web/20140730033501/http://www.geo.info.hu/sdila/prot/FOGALOM/Fogalom.htm#_Toc515006597
- Foote, K. D. (2023. 12 19). *Dataversity*. Forrás: dataversity.net: <https://www.dataversity.net/database-management-trends-in-2024/>
- Gál, T. (dátum nélk.). *Információ, adat, jel, kód*. Letöltés dátuma: 2024. 03 17, forrás: informatika.gtportal.eu: https://informatika.gtportal.eu/index.php?f0=alapfogalmak_01
- Garcia-Molina, H., D. Ullman, J., & Widom, J. (2008). *Database System The Complete Book*. New Jersey: Pearson Prentice Hall.
- Goodfellow, I., Bengio, Y., & Courville, A. (2015). *Deep Learning*. Cambridge, USA: The MIT Press.
- Hillyer, M. (2011. 06 06). *Managing Hierarchical Data in MySQL*. Letöltés dátuma: 2024. 04 09, forrás: web.archive.org: <https://web.archive.org/web/20110606032941/http://dev.mysql.com/tech-resources/articles/hierarchical-data.html>
- int, bigint, smallint, and tinyint (Transact-SQL)*. (2023. 05 23). Letöltés dátuma: 2024. 04 09, forrás: learn.microsoft.com: <https://learn.microsoft.com/en-us/sql/t-sql/data-types/int-bigint-smallint-and-tinyint-transact-sql?view=sql-server-ver16>
- J. Sadalage, P., & Fowler, M. (2013). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Crawforrdsville: Addison-Wesley.
- Kent, W. (1982. 09). *A Simple Guide to Five Normal Forms in Relational Database Theory*. California: Computing Machinery, Inc.
- Maurer, H., & Scherbakov, N. (2006. 09 04). *Network (CODASYL) Data Model*. Letöltés dátuma: 2024. 04 09, forrás: web.archive.org: <https://web.archive.org/web/20060904190944/http://coronet.iicm.edu/wbtmaster/allcours/escontent/netlib/ndm1.htm>

Murphy, K. P. (2012). *Machine Learning*. Cambridge, USA: The MIT Press.

Ramakrishnan, R., & Gehrke, J. (2002). *Database Management Systems*. New York: McGraw-Hill.

Transact-SQL reference (Database Engine). (2023. 07 13). Letöltés dátuma: 2024. 04 09, forrás: learn.microsoft.com: <https://learn.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15>

What is a cloud database? (2018. 03). Letöltés dátuma: 2024. 04 09, forrás: oracle.com: <https://www.oracle.com/database/what-is-a-cloud-database/>

What is Data Security? (2020. 12 06). Letöltés dátuma: 2024. 04 13, forrás: Oracle: <https://www.oracle.com/au/security/database-security/what-is-data-security/>

What is SQL Server Management Studio (SSMS)? (2023. 03 31). Letöltés dátuma: 2024. 04 17, forrás: Microsoft: <https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>

MELLÉKLETEK

NYILATKOZAT

Alulírott VARGA KRISTÓF büntetőjogi felelősségem tudatában nyilatkozom, hogy a szakdolgozatomban foglalt tények és adatok a valóságnak megfelelnek, és az abban leírtak a saját, önálló munkám eredményei.

A szakdolgozatban felhasznált adatokat a szerzői jogvédelem figyelembevételével alkalmaztam.

Ezen szakdolgozat semmilyen része nem került felhasználásra korábban oktatási intézmény más képzésén diplomaszerezés során.

Tudomásul veszem, hogy a szakdolgozatomat az intézmény plágiumellenőrzésnek veti alá.

Budapest, 2024. év 04 hónap 06 nap

Varga Kristóf

hallgató aláírása